82HM-000010-1

パーソナルコンピュータ

FM-7

ユーザーズマニュアル システム解説

富士通

パーソナルコンピュータ

FM-7

ユーザーズマニュアル システム解説

富士通株式会社

お　願　い

1．本書を始めとする各種マニュアルについてのお問合わせは，お買上げの販売店，および「富士通マイコンスカイラブ」へお願いいたします．
2．本体および各種オプション品，周辺装置の取扱いについては，各種取扱説明書を充分にお読みのうえ，使用して下さい．
3．新製品ニュース，ソフトウェアについてのお知らせは，各種マイコン雑誌への広告および，保証書，アンケート用紙をご送付くださった方へのダイレクトメール等により行います．

富士通マイコンスカイラブ

虎ノ門：〒106　東京都港区虎ノ門2－3－13　第18森ビル内
TEL（03）591－1091，2561
月～金（祝日を除く）　9時30分～17時

秋葉原：〒101　東京都千代田区外神田1－15－16　秋葉原ラジオ会館6F
TEL（03）251－1448
年中無休　10時～19時

札　幌：〒060　札幌市中央区南一条西3丁目　丸井今井一条本館4F
TEL（011）241－4185
月～金（水曜定休日）　10時～18時
土・日　10時～18時30分

仙　台：〒980　仙台市国分町1－7－18　明治生命仙台国分町ビル1F
TEL（0222）66－8711

名古屋：〒460　名古屋市中区栄1－5－22　富士通OAショールーム内
TEL（052）221－6016
月～土（祝日を除く）　10時～18時

大　阪：〒530　大阪市北区梅田1－2－2　大阪駅前第2ビル1F
TEL（06）344－7628
年中無休　10時～19時

広　島：〒733　広島市中区立町4－2　大橋ビル2F，3F
TEL（082）247－3949
年中無休　10時～19時

はじめに

このたびは，富士通パーソナルコンピュータFUJITSU MICRO 7（FM-7）をお買い求めいただきありがとうございました．

富士通では，すでに発売し，FM-8の愛称で多くの方々にご愛用いただいております，FUJITSU MICRO 8にFUJITSU MICRO 7およびFUJITSU MICRO 11を加え，FMシリーズとして富士通パーソナルコンピュータの充実をはかりました．

富士通が初めて製品化したFM-8は，それまでのパーソナルコンピュータと異なり，グラフィック機能の強化，大容量メモリの搭載，漢字キャラクタROM（オプション）の搭載による日本語処理機能をもたせるなど，機能を一段と向上させ，しかもコスト・パーフォーマンスの優れたものにいたしました．

このため，FM-8は本格的にビジネスユースとして適用できるものとなり，オフィスオートメーション（OA）化における中核的機器として位置づけが高くなってまいりました．もちろん，パーソナルユースとしても，分解能の高いカラーグラフィックが楽しめ，大量のデータが扱えるなど，また各種オプションの追加を容易にするI/Oインターフェースの豊富さは，応用範囲を極めて広くするものとして，将来の拡張性に楽しみのあるパーソナルコンピュータとして高い評価をいただいております．

さて，今回製品化いたしましたFM-7は，FMシリーズの中にあって，FM-8の下位に位置するものであります．

FM-7の製品化の意図は，これからますます発展していくパーソナルコンピュータ時代の中にあって，家庭やビジネスの分野において，手軽に入手でき，パーソナルコンピュータが身近かなものとして，ご利用いただけるものとしたところにあります．

本機の特徴のいくつかを挙げますと

・コスト・パフォーマンスに優れ，かつ低価格．
・グラフィック機能の充実．
・演算処理の高速化．
・大容量メモリの標準実装．
・漢字ROMカード（オプション）による日本語処理機能．
・目的別に選択できる豊富な周辺装置群．
・容易なシステムグレードアップ．

・プログラミング言語F-BASIC Version 3.0の実装.

・FM-8用ソフトウェアと完全互換.

・各種オペレーティングソフトウェア（OS）をサポート.

であります.

最新の半導体技術とパーソナルコンピュータ技術を最大限に生かしたFM-7は，ご使用してくださる方々に満足していただけるものと確信しております.

なにとぞ永い間，ご愛用くださいますようお願い申しあげます.

● 取扱い上のご注意 ●

・電　　源

・電源コードを充分，本体に差しこんで下さい．

・電源はAC100V(商用周波数50Hz／60Hz) を使用して下さい．

・電源コードは，必ず本体の電源スイッチを「OFF」にしたあと，電源プラグをつかみ，抜き差しして下さい．

・周辺装置類を接続，切離しの時は，必ず本体の電源スイッチを「OFF」にして下さい．

・本体背面のディップスイッチを切り換える際は，必ず電源を「OFF」にした状態にて行なって下さい．

・システムを起動する時は，周辺装置から電源スイッチを「ON」にし，本体の電源スイッチは最後に「ON」にして下さい．

・電源スイッチを「ON」の状態から「OFF」にし，再び「ON」状態にする時は，5秒以上の間隔をとって下さい．誤まって電源からプラグが外れた時も同様です．

・設置および保管

・本体は精密な電子部品を使用していますので，衝撃を加えたり，振動の加わる場所はさけて下さい．

・極端に高い温度や低い温度，温度変化の大きい場所，直射日光の当る場所，発熱する機器の近くはさけて下さい．

・湿度の高い場所，ほこりの多い場所，薬品類の雰囲気中はさけて下さい．

・本体には温度上昇を抑えるため，通気孔を設けています．通気孔をシートなどで覆ったり，本体を通気の悪い狭い場所に押しこんだりしての使用はさけて下さい

・水や液体および金属類が入った状態での使用は危険ですし，また故障の原因にもなります．異物が入らないように注意して下さい．

・テレビやラジオとあまり近い場所で使用しますと，テレビやラジオにノイズとして悪い影響を及ぼす場合があります．また，強い磁界を発生する機器の近くで使用しますと，ノイズとして逆に影響を受ける場合があります．これらの場合は本体を移動して下さい．

・本体を解体した状態での保管および使用は危険ですし，また故障の原因にもなりますのでおやめ下さい．

・そ の 他

・本体についたほこりを落す場合は，やわらかい布状のものを使用して下さい．汚れを落す場合は，水や洗剤を含ませた布で軽くふきとって下さい．揮発性の薬品類はさけて下さい．

・本体の内部にほこりがたまってしまった場合は，部品類にふれないよう掃除機で吸いとるか，風を吹きかけてとり除いて下さい．

万一，故障や異常に気づきましたら，本体を電源から外し，お買い求めいただいた販売店，もしくは，もよりの富士通マイコン・スカイラブにご相談下さい．

目　次

第1章　概　要

1.1　ハードウェア ……………………………………………… 1 - 1

1.1.1　特　長 ……………………………………………… 1 - 1

1.1.2　本体ブロック図 ……………………………………… 1 - 3

1.2　ソフトウェア ……………………………………………… 1 - 4

第2章　本体および周辺装置

2.1　FM-7（MB25010） ………………………………………… 2 - 1

2.2　周辺装置 ……………………………………………… 2 - 2

2.2.1　CRTディスプレイ …………………………………… 2 - 2

2.2.2　補助記憶装置 ……………………………………… 2 - 3

2.2.3　プリンタ ……………………………………………… 2 - 5

2.3　システム構成 ……………………………………………… 2 - 6

2.3.1　最小システム構成 ………………………………… 2 - 6

2.3.2　標準システム構成 ………………………………… 2 - 6

2.3.3　拡張システム構成 ………………………………… 2 - 7

第3章　本体と周辺装置の接続および動作確認

3.1　はじめに ……………………………………………… 3 - 1

3.2　FM-7の本体 ……………………………………………… 3 - 2

3.2.1　FM-7の外観 ……………………………………… 3 - 2

3.2.2　本体各部の説明 …………………………………… 3 - 3

3.2.3　電源ケーブルの接続 ……………………………… 3 - 4

3.2.4　電源スイッチの投入 ……………………………… 3 - 4

3.2.5　本体の背面 ………………………………………… 3 - 5

3.3　CRTディスプレイ ………………………………………… 3 - 6

3.3.1　カラーCRTディスプレイの接続 …………………… 3 - 7

3.3.2 グリーンCRTディスプレイの接続 ……………………………… 3-8
3.3.3 家庭用テレビの接続 ……………………………………………… 3-8
3.3.4 CRTディスプレイのテスト ……………………………………… 3-11
3.4 プリンタ ……………………………………………………………… 3-13
3.4.1 シリアルドットプリンタの接続 ………………………………… 3-14
3.4.2 シリアルドットプリンタのテスト ……………………………… 3-15
3.5 カセットテープレコーダ ……………………………………………… 3-16
3.5.1 FMデータレコーダの接続 ………………………………………… 3-16
3.5.2 FMデータレコーダのテスト ……………………………………… 3-17
3.6 ミニフロッピィディスクユニット …………………………………… 3-18
3.6.1 ミニフロッピィインタフェースカードの装着 ………………… 3-19
3.6.2 ミニフロッピィアダプタの接続 ………………………………… 3-21
3.6.3 ミニフロッピィディスクユニットのテスト …………………… 3-22
3.6.4 ミニフロッピィディスク取扱上の注意事項および仕様 ……… 3-24
3.7 オプションカード ……………………………………………………… 3-27

第4章 システムの起動

4.1 システムの起動 ……………………………………………………… 4-1
4.1.1 ディップスイッチ ………………………………………………… 4-1
4.1.2 ブザーボリューム，サウンドボリュームの調整 ……………… 4-2
4.2 F-BASICの起動 ……………………………………………………… 4-3
4.2.1 F-BASIC (DISKモード)の起動 ………………………………… 4-4
4.2.2 F-BASIC (ROMモード)の起動 ………………………………… 4-6
4.3 FM-7で扱えるDOS …………………………………………………… 4-7

第5章 キーボード

5.1 キーボード概要 ……………………………………………………… 5-2
5.1.1 英文字モード ……………………………………………………… 5-3
5.1.2 グラフィックモード ……………………………………………… 5-4
5.1.3 カナモード ………………………………………………………… 5-5
5.2 特殊キー ……………………………………………………………… 5-6
5.3 エディットキー ………………………………………………………… 5-11
5.4 プログラマブルファンクションキー ………………………………… 5-13

5.5　テ ン キ ー ……………………………… 5-14
5.6　カナキーの位置 ……………………………… 5-15

第6章　プログラムの入力と修正方法

6.1　プログラムの入力方法 ……………………………… 6-1
6.1.1　AUTO (オート) コマンドによる方法 ……………………………… 6-1
6.1.2　行番号入力による方法 ……………………………… 6-4
6.1.3　プログラムの実行 ……………………………… 6-4
6.1.4　プログラムリストの表示 ……………………………… 6-5
6.2　プログラムの修正方法 ……………………………… 6-6
6.2.1　カーソル移動キー ……………………………… 6-6
6.2.2　文字の訂正 ……………………………… 6-8
6.2.3　文字の挿入 ……………………………… 6-9
6.2.4　文字の削除 ……………………………… 6-11
6.2.5　文 の 挿 入 ……………………………… 6-12
6.2.6　文 の 削 除 ……………………………… 6-14
6.3　そ の 他 ……………………………… 6-16
6.3.1　[ESC] キー ……………………………… 6-16
6.3.2　[PF10] ～ [PF1] の設定 ……………………………… 6-16

第7章　ディスプレイ表示

7.1　キャラクタ表示 ……………………………… 7-1
7.1.1　FM-7のキャラクタ座標 ……………………………… 7-1
7.1.2　表示文字数の特長 ……………………………… 7-1
7.1.3　各モードによるディスプレイの状態 ……………………………… 7-2
7.1.4　表示文字数の変更 (WIDTH 命令) ……………………………… 7-2
7.1.5　キャラクタ表示 ……………………………… 7-2
7.2　グラフィック表示 ……………………………… 7-3
7.2.1　FM-7のグラフィック座標 ……………………………… 7-3
7.2.2　グラフィック表示 ……………………………… 7-4
7.3　単色3画面表示 ……………………………… 7-4
7.3.1　概　　要 ……………………………… 7-4
7.3.2　スクリーン命令の定義 ……………………………… 7-4

7.4 パレット機能 …… 7-14
7.4.1 概　　要 …… 7-14
7.4.2 パレットコードの定義 …… 7-14
7.4.3 COLOR文の動作 …… 7-15
7.5 プログラム例 …… 7-16

第8章　音楽演奏機能

8.1 概　　要 …… 8-1
8.2 PLAY文の使い方 …… 8-1
8.2.1 音量（Vn） …… 8-1
8.2.2 テンポ（Tn） …… 8-1
8.2.3 休符（Rn） …… 8-2
8.2.4 音程（A, B, C, D, E, F, G） …… 8-2
8.2.5 オクターブ（On） …… 8-2
8.2.6 音長（Ln） …… 8-2
8.2.7 特殊音程（Nn） …… 8-3
8.2.8 エンベロープパターン（Sn） …… 8-4
8.2.9 エンベロープ周波数（Mn） …… 8-4
8.3 PLAY文による和音 …… 8-5
8.4 SOUND文の使い方 …… 8-6
8.4.1 PSGの概要 …… 8-6
8.4.2 レジスタの説明 …… 8-7
8.4.3 SOUND文の使用例 …… 8-8

第9章　日本語表示

9.1 漢字の画面への表示方法 …… 9-1
9.2 漢字を1文字表示する場合 …… 9-2
9.3 漢字を1行へ複数表示する場合 …… 9-2
9.4 漢字を複数行表示する場合 …… 9-3
9.5 漢字コードをデータとして扱う方法 …… 9-3
9.6 漢字をスクロールさせ表示する方法 …… 9-4

第10章　その他の機能

10.1　ブートローダ機能 ………………………………………… 10-1

10.2　動作クロック周波数の設定 ………………………………… 10-2

10.3　ウォームスタート ………………………………………… 10-2

10.4　プログラム作成上の注意 ………………………………… 10-3

第11章　ファイル

11.1　ファイルの概念 …………………………………………… 11-1

11.2　ファイルディスクリプタ ………………………………… 11-2

　11.2.1　ファイル名 ……………………………………………… 11-3

　11.2.2　デバイス名 ……………………………………………… 11-3

11.3　ファイルの扱い（ファイルの種類） ……………………… 11-4

　11.3.1　プログラムファイル（ロード・セーブ） ……………… 11-5

　11.3.2　データファイル ………………………………………… 11-6

第12章　カセットテープ

12.1　テープレコーダの扱い方 ………………………………… 12-1

　12.1.1　家庭用テープレコーダに対する調整法 ………………… 12-1

　12.1.2　FMデータレコーダに対する扱い ……………………… 12-3

12.2　リモートコントロール機能 ……………………………… 12-4

12.3　ファイル名の出力方法 …………………………………… 12-4

12.4　プログラムファイルの扱い方 …………………………… 12-5

　12.4.1　LOADとSAVE …………………………………………… 12-5

　12.4.2　LOADMとSAVEM ……………………………………… 12-8

　12.4.3　ロード，セーブに関する注意事項 ……………………… 12-10

12.5　チェックサムの照合 ……………………………………… 12-11

12.6　データファイルの扱い方 ………………………………… 12-12

　12.6.1　データファイルの作成 ………………………………… 12-12

　12.6.2　データファイルからのデータ入力 ……………………… 12-22

12.7　MERGE命令とCHIN命令の使い方 ……………………… 12-29

　12.7.1　MERGE（マージ） ……………………………………… 12-29

　12.7.2　CHAIN（チェイン）・COMMON（コモン） ……………… 12-31

12.8　カセットテープにおけるデータの記録形式 ……………… 12-36

第13章　ミニフロッピィディスク

13.1　ファイルディスクリプタ………… 13-1
13.2　ファイル名の出力方法(FILES)………… 13-2
13.3　ファイル名の変更(NAME)………… 13-3
13.4　ファイルの消去………… 13-4
13.4.1　すべてのファイルの消去………… 13-4
13.4.2　指定したファイルの消去………… 13-5
13.5　プログラムファイルの扱い方………… 13-6
13.5.1　LOADとSAVE………… 13-6
13.5.2　LOADMとSAVEM………… 13-7
13.6　データファイルの扱い方………… 13-9
13.6.1　シーケンシャルファイルの扱い方………… 13-9
13.6.2　ランダムファイルの扱い方………… 13-22
13.7　MERGE命令とCHAIN命令の使い方………… 13-33
13.7.1　MERGE(マージ)………… 13-33
13.7.2　CHAIN(チェイン)・COMMON(コモン)………… 13-34
13.8　ミニフロッピィディスクに関する特殊な命令………… 13-37
13.8.1　DSKO$(ディスク・オー・ダラー)………… 13-37
13.8.2　DSKI$(ディスク・アイ・ダラー)………… 13-37
13.8.3　DSKF(ディスク・エフ)………… 13-38
13.9　ファイルの構成と管理………… 13-38
13.9.1　ミニフロッピィディスクの構造………… 13-38
13.9.2　セクタアドレス………… 13-39
13.9.3　クラスタ………… 13-40
13.9.4　トラックの割当て………… 13-40
13.9.5　トラック0と1の構成………… 13-41
13.9.6　FAT(ファット)………… 13-41
13.9.7　ID(アイ・ディー)………… 13-42
13.9.8　ディレクトリ………… 13-42
13.10　ミニフロッピィディスクの破壊………… 13-43
13.10.1　使用環境によるもの………… 13-43
13.10.2　使用方法によるもの………… 13-43
13.11　ディスクユーティリティ………… 13-44

13.11.1　システムディスクの作成 ……13-45
13.11.2　ミニフロッピィディスクのコピー ……13-51
13.11.3　補助プログラム ……13-54
13.11.4　プログラムの自動スタート ……13-55
13.11.5　PF キーの初期設定 ……13-56

第14章　プリンタ

14.1　バッファとカウンタ……14-1
14.2　F-BASICにおけるプリンタに対する命令……14-1
14.3　プログラムリストの出力……14-3
14.4　プリンタへのデータ出力……14-5
14.5　OPEN命令における指定 ……14-8
14.6　コントロールコードによるプリンタの制御……14-9
14.6.1　コマンド……14-13
14.6.2　エスケープコントロール……14-17
14.7　漢字の出力方法……14-34
14.7.1　ドットプリンタおよびシリアルドットプリンタに対する出力方法……14-35
14.7.2　ビジネスプリンタに対する出力方法……14-39

第15章　回線制御

15.1　概　　要……15-1
15.2　インタフェース……15-2
15.3　ディップスイッチの設定方法……15-3
15.4　ボーレートの選択……15-4
15.5　F-BASICにおけるエラー発生表示……15-5

付　録

1　キーボード配列図 ……付録-2
2　キャラクタコード表……付録-4
3　非漢字一覧表 ……付録-5
4　JIS 第1水準漢字一覧表 ……付録-7
5　F-BASIC エラーリスト ……付録-18
6　機器接続図 ……巻末

第1章　概　　要

FUJITSU MICRO 7 (FM-7) は，パーソナルコンピュータの本体とキーボードがコンパクトに一体化してあります．

これらを構成する電子部品には，CPUやFM-7専用に開発した各種ロジック用LSI，大容量メモリLSI，高品質で操作性の優れたキーボードなどを採用し，高性能，高信頼度，コスト・パフォーマンスの優れたパーソナルコンピュータが実現しました．

プログラム言語は，標準としてF-BASIC Version3.0 (V3.0) ですが，FM-8用F-BASICのプログラムが，そのまま適用できるように設計してあります．また，ディスクオペレーティングシステム（DOS）では，UCSD Pascal™，FLEX™も扱え流通している各種言語，アプリケーションソフトウェアが使用できます．

FMシリーズには，各種周辺装置が数多く用意してありますので，用途に合せて，これらの中から選ぶことにより，効果的なシステムの構成ができます．システムを段階的に拡張していく場合も考慮して，お選び下さい．

1.1　ハードウェア

1.1.1　特　　長

FM-7のハードウェアの主なる特長は次の通りです．

- **究極の8ビットCPU MBL6809を2個搭載**（8 MHz/4.9 MHz切換え可）

  メインCPU　　MBL68B09

  サブCPU　　MBL68B09（ディスプレイ，キーボード制御用など）

- **高集積度半導体メモリの採用**

  RAM　64KB　　ユーザエリア（64K bit DRAM採用）

  ROM　32KB　　F-BASIC　V3.0（256K bit CMOS　ROM採用）

  RAM　48KB　　ビデオRAM（16K bit DRAM採用）

  その他，ブートローダ，CRTモニタ，キャラクタパターンなど．

- **性能アップに専用に開発したLSIを採用**

  CMOS　2000ゲートアレー　　TTL　200ゲートアレー

  TTL　500ゲートアレー

・漢字ROMカード（オプション）で日本語処理機能

ROM　128 KB　　漢字2965字/非漢字453字

16ドット×16ドット構成（256 Kbit CMOS ROM採用）

・操作性の優れたキーボード

シリンドリカル・ステップ・スカルプチャ構造

4 bit マイクロコンピュータによるキースキャン制御，キー数　98個

・高解像度ディスプレイ機能

画面構成　80字×25行（2000文字）

40字×20行（800文字/家庭用テレビ使用時）

文字構成　8ドット×8ドット

カラー　8色（パレット機能で色交換が容易）

グラフィック　640ドット×200ドット，ドットごとに色指定可能，文字との混在可能

・単色3画面表示

・三和音のサウンド機能

・各種インタフェースを標準実装

オーディオカセットインタフェース

プリンタインタフェース

CRTインタフェース

外部スピーカポート

I/O拡張ポート

・オプションカードの本体内装着用スロット（3枚分）

1.1.2 本体ブロック図

メイン部
CPU
MBL68B09
ROM
ブートローダ
4KB
ROM
F-BASIC V3.0
32KB
RAM
ユーザエリア
64KB
PSG
スピーカ
オーディオカセット
インタフェース
プリンタ
インタフェース
I/O 拡張ポート
オプションカード用
スロット
オプションカード用
スロット
オプションカード用
スロット
CPU間
インタフェース
サブ部
CPU
MBL68B09
ROM
モニタ
8KB
ROM
キャラクタジェ
ネレータ 2KB
RAM
コンソール処理
4KB
RAM
ビデオ RAM
48KB
CRT
インタフェース
キーボード
オーディオカセット
コネクタ
オーディオ
カセット
プリンタコネクタ
プリンタ
I/O 拡張コネクタ
I/O 拡張ユニット
(3枚まで収納できます)
ミニフロッピィ
インタフェースカード
ミニフロッピィ
ディスク
RS-232C
インタフェースカード
音響カプラ等
漢字 ROM カード
Z80 カード
カラー CRT
コネクタ
カラーディスプレイ
家庭用テレビ
アダプタ
家庭用テレビ
グリーンディスプレイ
グリーン CRT
コネクタ

1.2 ソフトウェア

FM-7のプログラム言語は，F-BASIC V3.0が標準となり，32KBをマスクROM化し，本体内に標準実装しております．

F-BASIC V3.0はマイクロソフト社のBASICを強化したFM-8用F-BASICが基になっております．

F-BASIC V3.0の大きな特長は，FM-7とFM-8が異なる次の機能をサポートするコマンドやステートメントを追加，もしくは削除したことを除けば，F-BASICと完全に互換性をもたせたことにあります．

追加	パレット機能
	単色3画面表示
	サウンド機能
削除	バブルカセットメモリに対する命令

したがって，FM-8用に流通しているソフトウェアが，そのままFM-7に適用できます（ただしバブルカセットによりサポートしているソフトウェアは除きます）．

このように，FM-7では，FM-8や各種パーソナルコンピュータ用に流通しているアプリケーションプログラムの中から，目的に合ったソフトウェアを選ぶことができますので，幅広い分野に対応することができます．

第２章　本体および周辺装置

本章では，本体の FM-7 (MB25010) とシステム構成に必要な各種周辺装置類を簡単に紹介します．

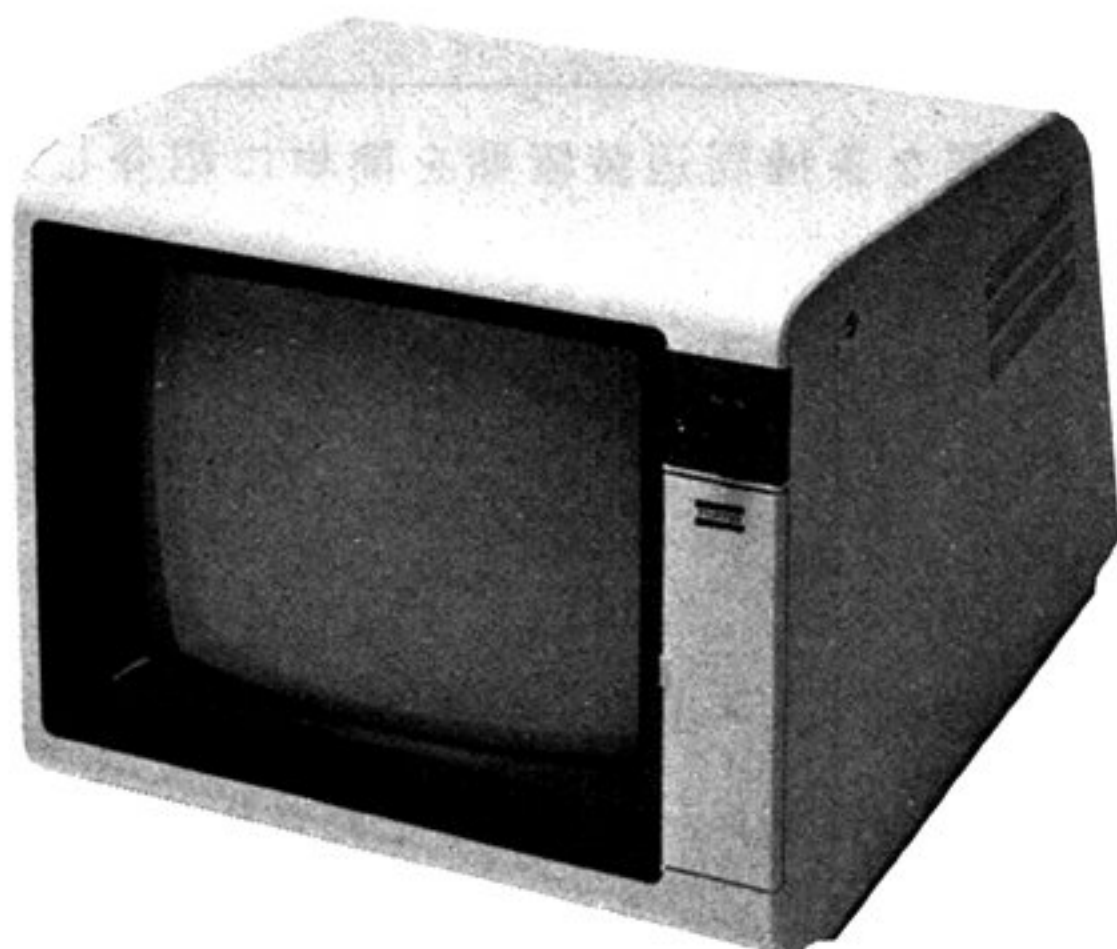
2.1 FM-7 (MB25010)

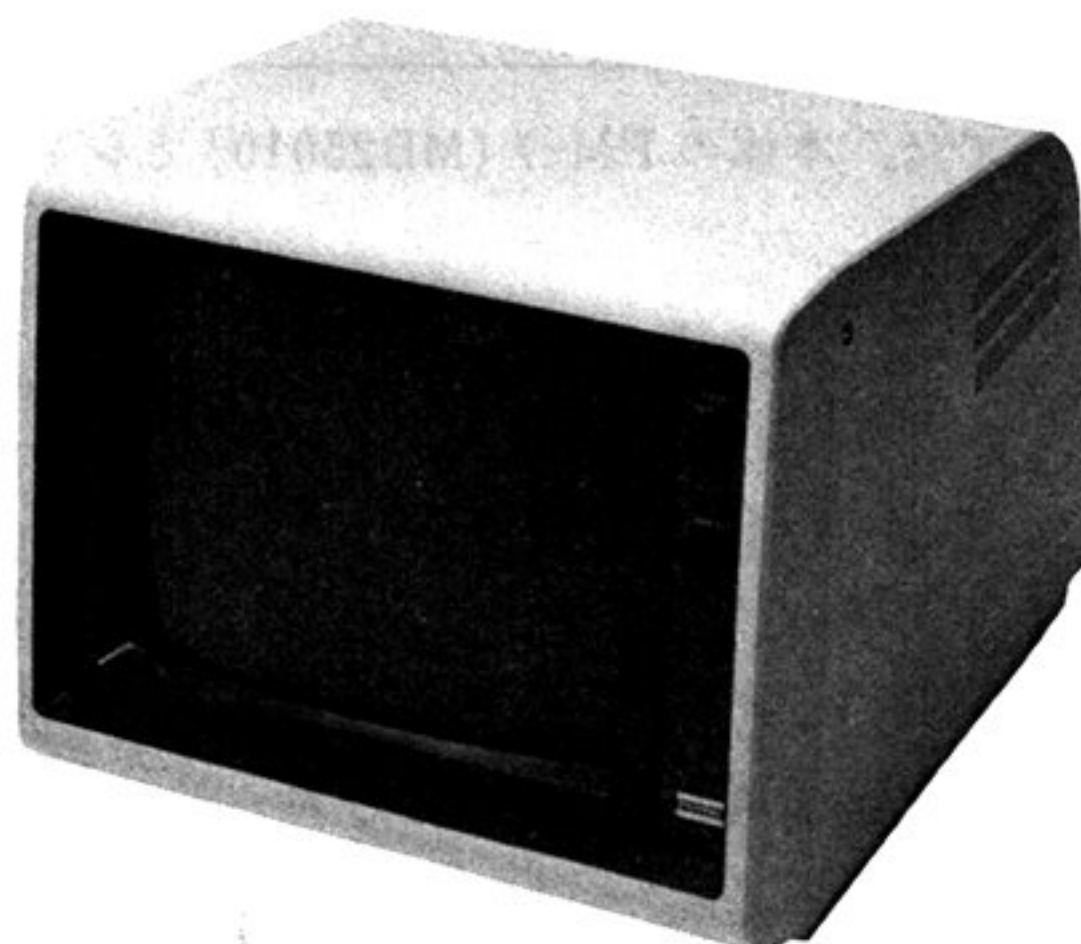
FM-7は，コンピュータとして必要な演算部，記憶部，制御部，入・出力インタフェース，オプションカード装着用スロットおよびキーボードが１つのコンソールにコンパクトにまとめられております．

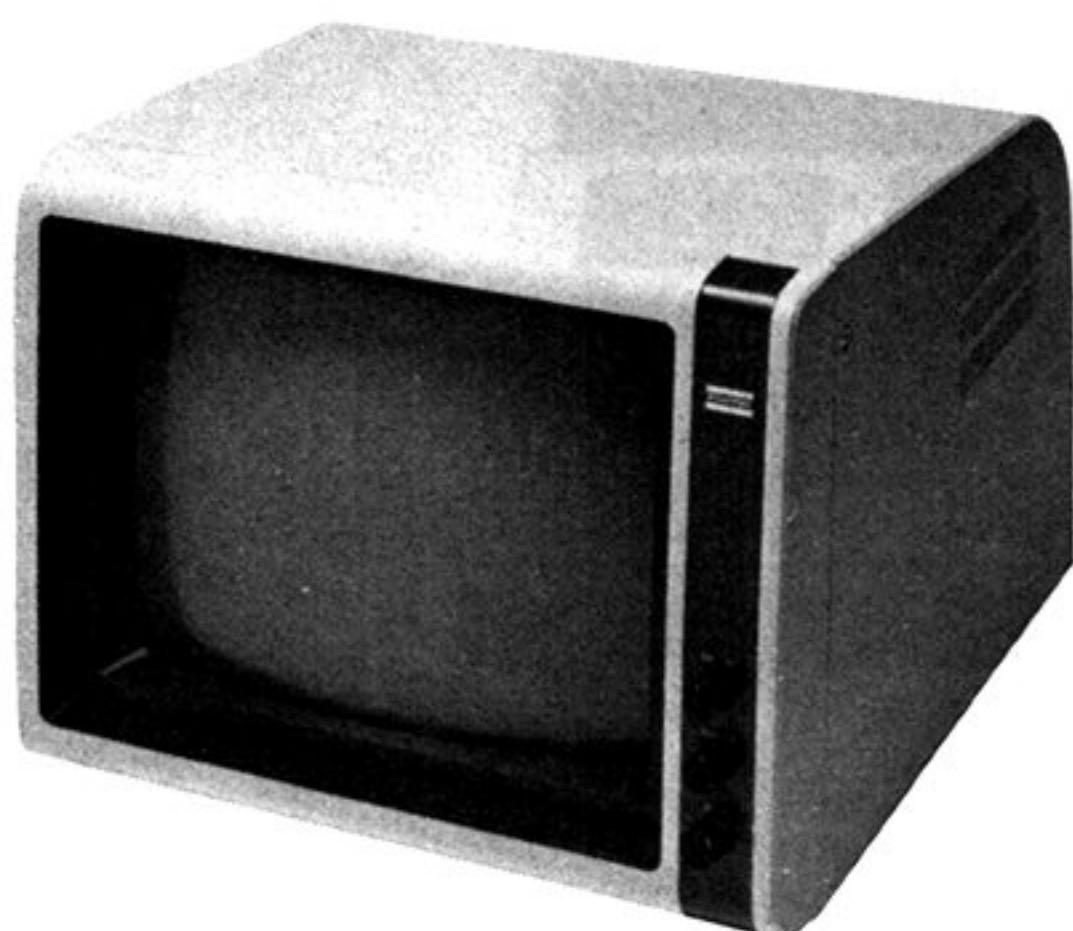
FM-7は，CPU，メモリなど最新のLSIに加え，FM-7専用に開発したLSIを採用するなど，小形で性能の高い，信頼性の高いパーソナルコンピュータです．

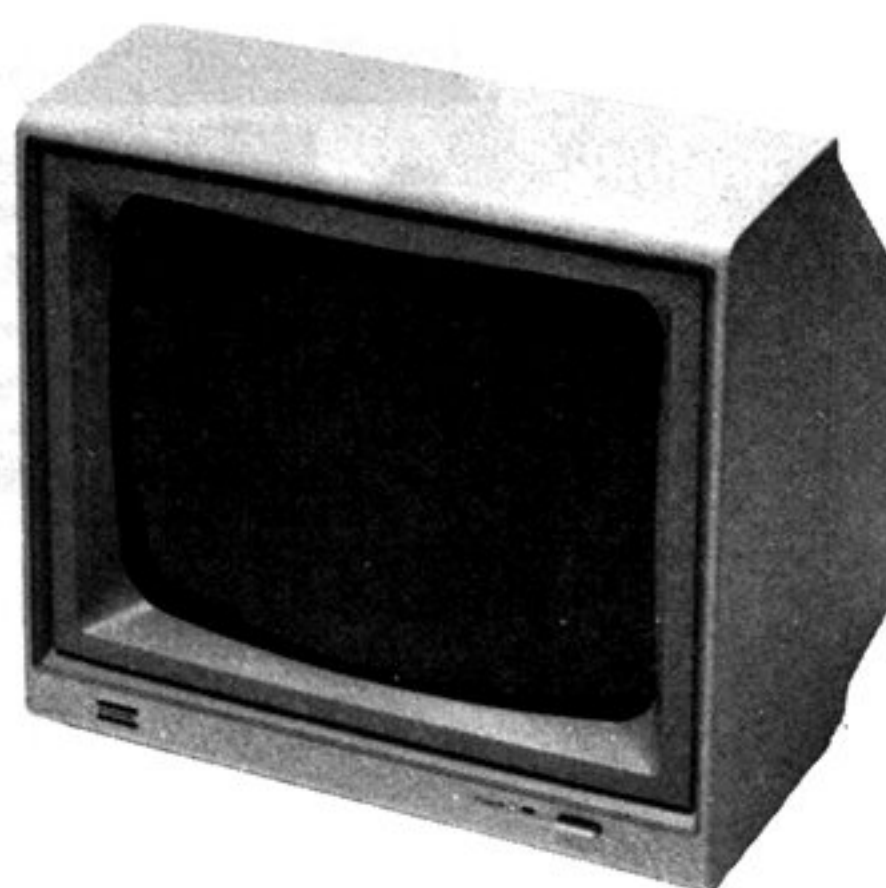
用途に合せて，FMシリーズの周辺装置を接続してシステムを構成できますので，経済的で拡張性のあるパーソナルコンピュータといえます．

2.2 周 辺 装 置

2.2.1 CRTディスプレイ

① 高解像度カラーCRTディスプレイ (MB27301)

ブラウン管	12インチ
表示面積	220×139 mm
解像度	640×200 ドット
色	8色

② カラーCRTディスプレイ (MB27303)

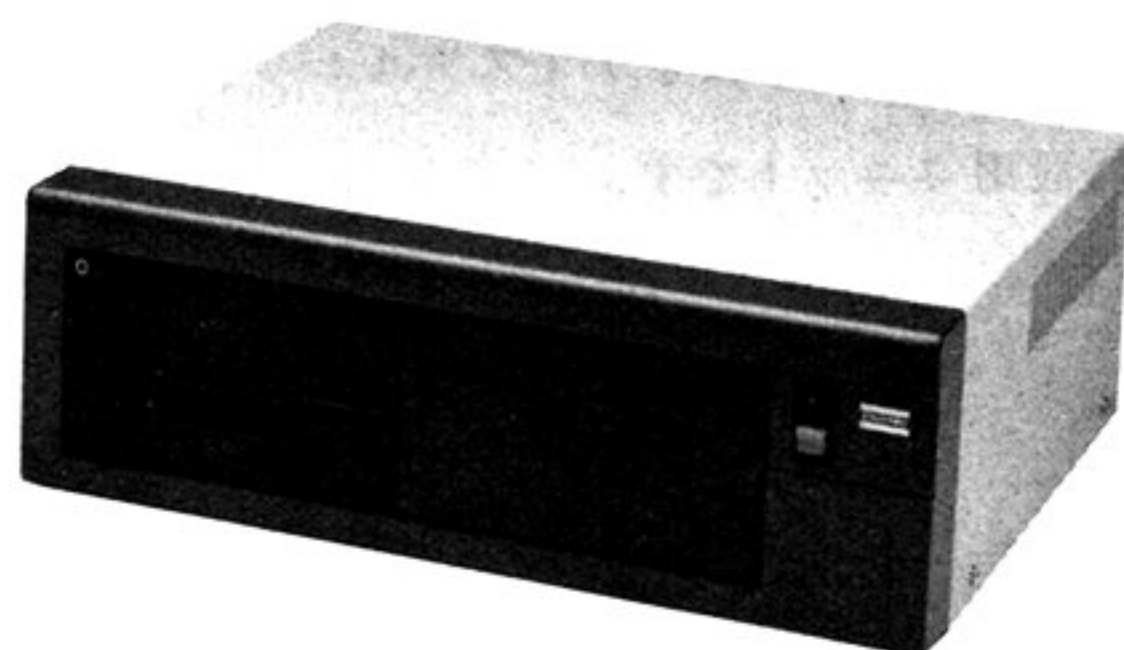
ブラウン管	12インチ
表示面積	220×139 mm
解像度	640×200 ドット
色	8色

③ 高解像度グリーンCRTディスプレイ (MB27302)

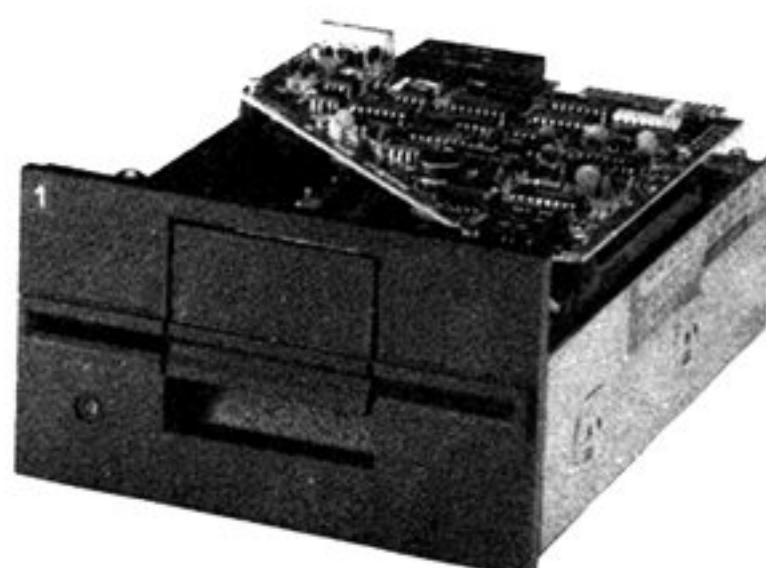
ブラウン管	12インチ
表示面積	220×139 mm
解像度	640×200 ドット
色	8階調

④ グリーンCRTディスプレイ (MB27304)

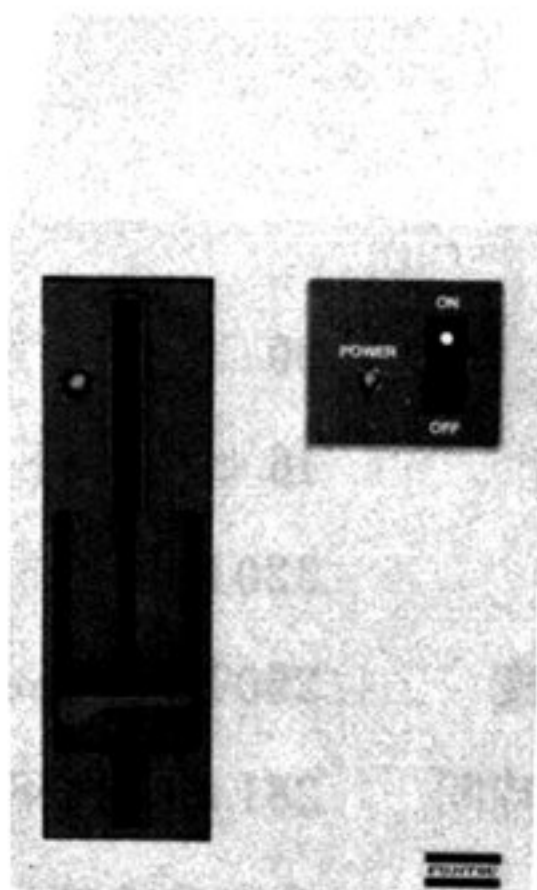
ブラウン管	12インチ
表示面積	220×139 mm
解像度	640×200 ドット
色	8階調

2.2.2 補助記憶装置

トラック数	80
セクタ数	16セクタ/トラック
記憶容量	320Kバイト×2
データ転送速度	250Kbit/sec
平均アクセス時間	281msec(平均)
ドライブ数	2ドライブ

⑤ ミニフロッピィディスクユニット (MB27601)

トラック数	80
セクタ数	16セクタ/トラック
記憶容量	320Kバイト
データ転送速度	250Kbit/sec
平均アクセス時間	281msec(平均)
ドライブ数	1ドライブ

⑥ Sタイプミニ FD ユニット (MB27605)

Sタイプミニフロッピィディスクユニットの増設用ドライブです.

⑦ ミニフロッピィディスクドライブ (MB27606)

トラック数	80
セクタ数	16セクタ/トラック
記憶容量	320Kバイト
データ転送速度	250Kbit/sec
平均アクセス時間	281msec(平均)
ドライブ数	1ドライブ

⑧ 薄形ミニFDユニット(MB27607)

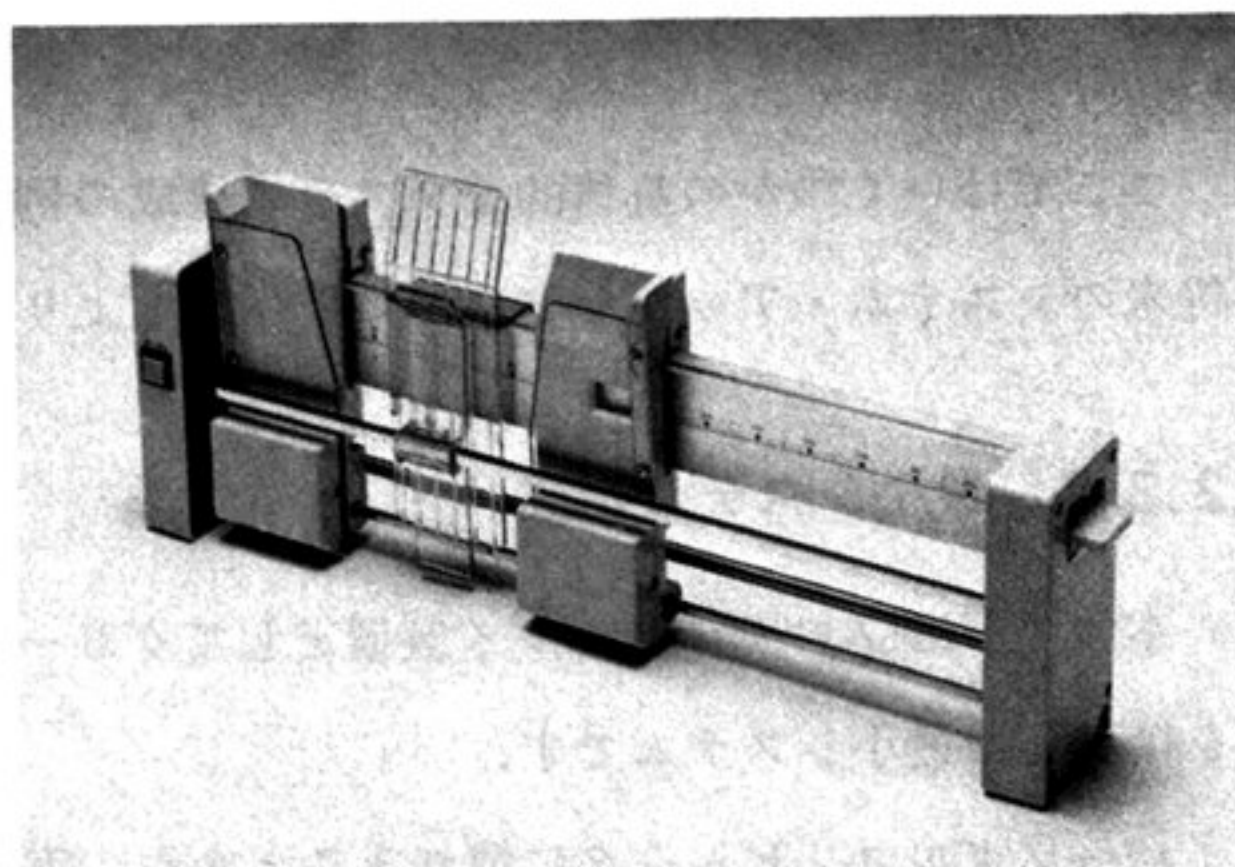
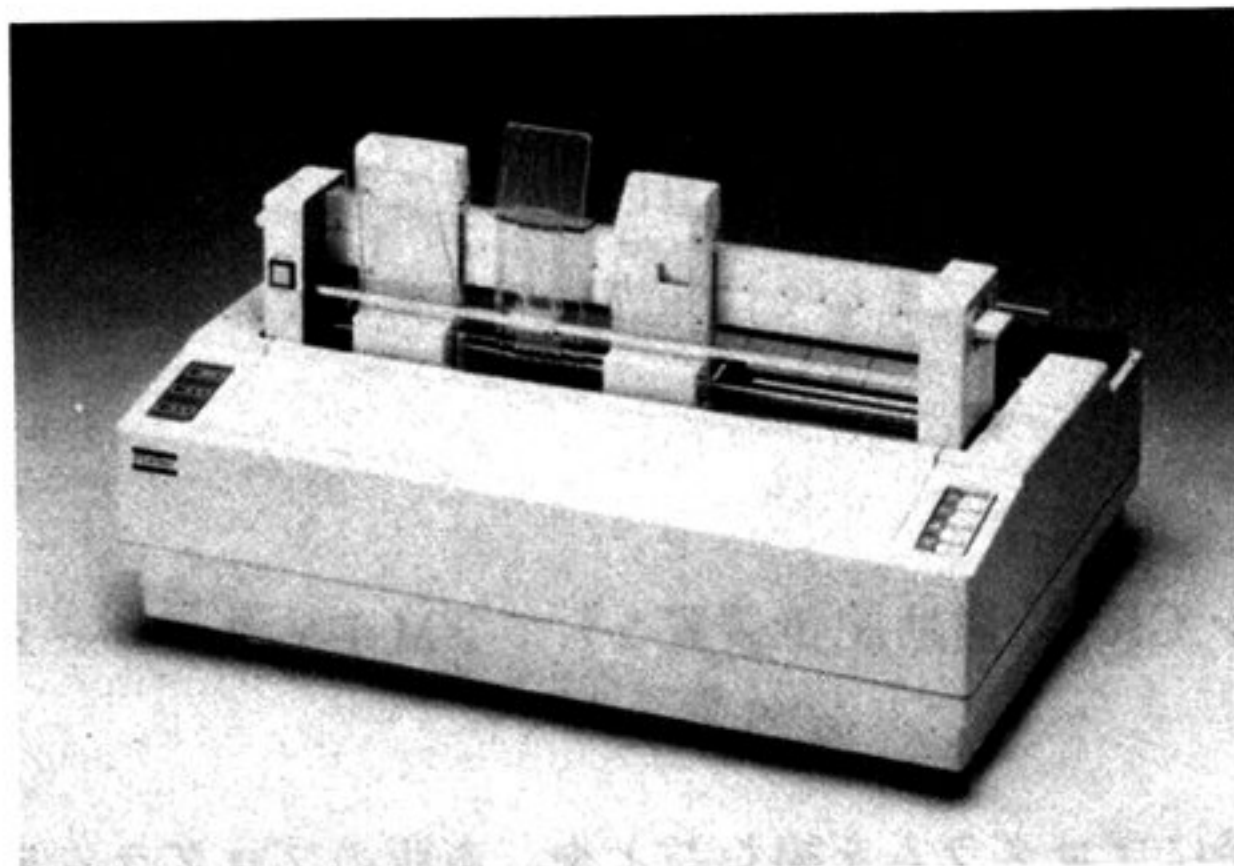
薄形ミニフロッピィディスクユニットの増設用ユニットです.

⑨ 増設用薄形ミニFDユニット(MB27608)

トラック方式	2トラック1チャンネル モノラル
スピーカ	66mm
周波数	250~6000Hz
電　源	AC100V/DC6V

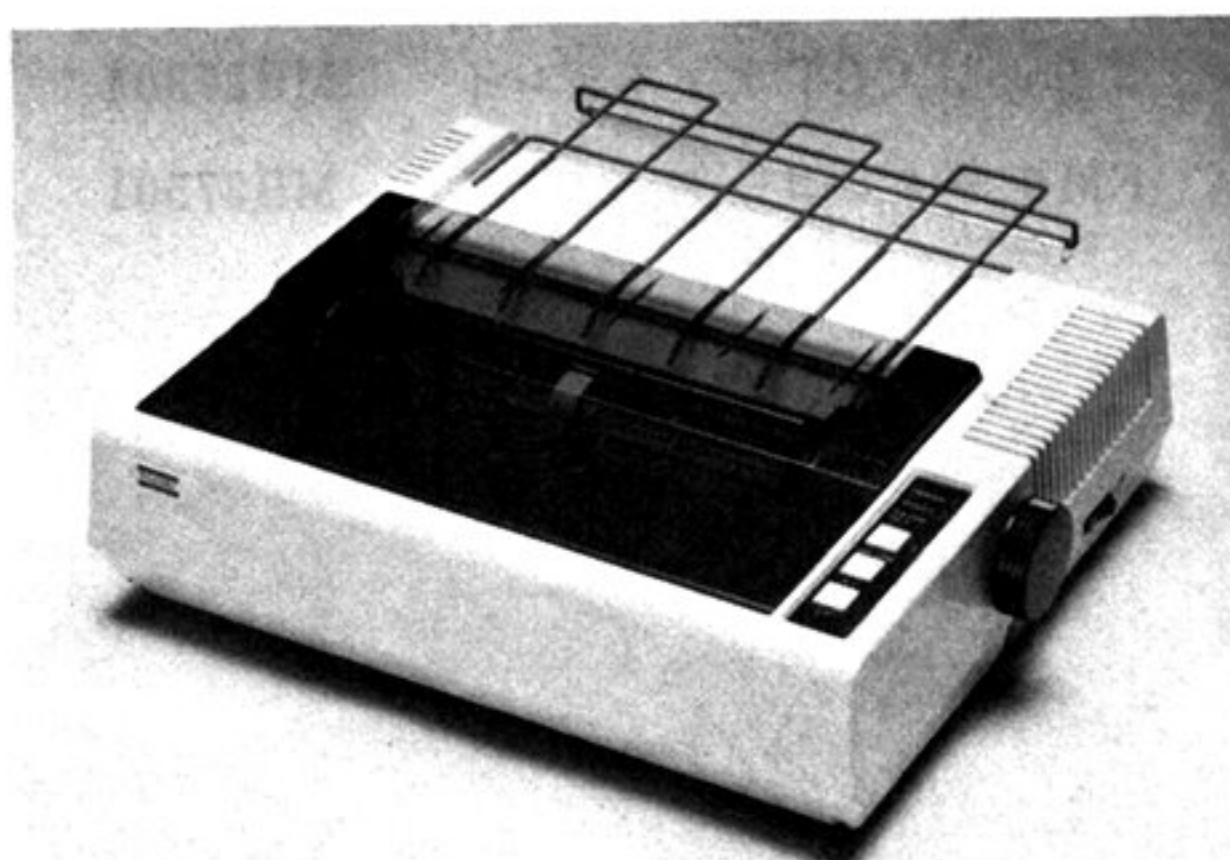
キュー・レビュー付

⑩ FMデータレコーダ(MB27501)

2.2.3 プ リ ン タ

⑪ ビジネスプリンタ（MB27402）

印字方式	シリアルドットマトリクス
印字桁数	136字（パイカピッチ） 90字（漢字）/行
印字速度	ANK 165字/sec 漢字 55字/sec
ピ ン	16ピン 千鳥配列
印字方向	両方向最短距離印字
コピー	5枚

⑫ フロントインサータ（MB26201）

定形印刷の用紙（単票）をセットして印字することができます．

⑬ シリアルドットプリンタ（MB27403）

印字方式	ドットインパクト
印字桁数	80字/行
印字速度	80字/sec
ピ ン	8ピン
印字方向	両方向
コピー	3枚

⑭ ドットプリンタ（MB27404）

印字方式	インパクトドットマトリクス
印字桁数	80字/行
印字速度	25字/sec
ピ ン	8ピン
印字方向	両方向
コピー	2枚

2.3　システム構成

ここではいくつかの具体的なシステム構成の例を紹介します．個々の目的に合せたシステム構成，将来のシステム・アップなどについては，もよりのスカイラブもしくは販売店でご相談ください．

2.3.1　最小システム構成

本体のFM-7にディスプレイ装置としてグリーンCRT，補助記憶装置としてFMデータレコーダを接続した最小システムです．

パーソナルコンピュータに慣れることから，実際にプログラムを組むことや，市販のプログラムを実行することができます．プログラムや，ディスプレイのハードコピーを必要としない場合は，このシステムで充分です．

カラーグラフィックが必要な場合は，カラーCRTを用いて下さい．また，解像度が必要でない場合は，家庭用テレビを用いても良いでしょう．この場合家庭用カラーテレビアダプタが必要です．

本　体	MB25010
グリーンCRTディスプレイ	MB27304
FMデータレコーダ	MB27501

2.3.2　標準システム構成

本体のFM-7に，ディスプレイ装置としてカラーCRTディスプレイ，補助記憶装置としてミニフロッピィディスクユニット，それにドットプリンタを接続しました．ビジネス用としての標準構成ですが，ホーム用としては高級システム構成といえます．

ミニフロッピィディスクユニットの使用により，大量のデータが扱えますし，プリンタの接続により，プログラムのリスト出力が可能となり，プログラムの本格的な開発，プログラム実行結果の記録ができます．

本　体	MB25010	ミニフロッピィディスクユニット	MB27601
カラーCRTディスプレイ	MB27303	ドットプリンタ	MB27404

2.3.3 拡張システム構成

本体のFM-7にディスプレイ装置として，カラーCRTディスプレイ，補助記憶装置としてミニフロッピィディスクユニット，プリンタには高性能なビジネスプリンタを接続しました．ビジネス用としてのシステム構成です．さらに漢字ROMカードを装着しますと，日本語処理もでき，本格的なOA機器として多用途に対応できます．

本　体	MB25010	ビジネスプリンタ	MB27402
高解像度カラーCRT ディスプレイ	MB27301	フロントインサータ	MB26201
ミニフロッピィディスクユニット	MB27601	漢字ROMカード	MB22405

第３章　本体と周辺装置の接続および動作確認

3.1 はじめに

この章では，本体と各周辺装置との接続方法を中心に，各部の名称，その他注意事項を書いてあります．実際にシステムを接続する前に必ずお読み下さい．

あなたの持っているFM-7のシステムを頭の中でならべて見て下さい．つぎにFM-7のシステムを置く場所を確保してみて下さい．そして，その場所で，接続作業ができるかどうか考えてみて下さい．

以上のことをしっかり確認してみて下さい．

3.2 FM-7の本体

3.2.1 FM-7の外観

<準　備>

箱をあけて下さい．中には次に示すものが入っていますので，御確認下さい．

a……電源ケーブル，オーディオカセットケーブル

b……グラフィックパターン用シール

c……マニュアル（ユーザーズマニュアルF-BASIC入門，F-BASIC文法書，ユーザーズマニュアルシステム解説，ユーザーズマニュアルシステム仕様，F-BASIC文法書，ポケットブック，NEW VIP操作マニュアル）

d……デモテープ（カセットテープ），NEW VIP（カセットテープ）

e……保証書

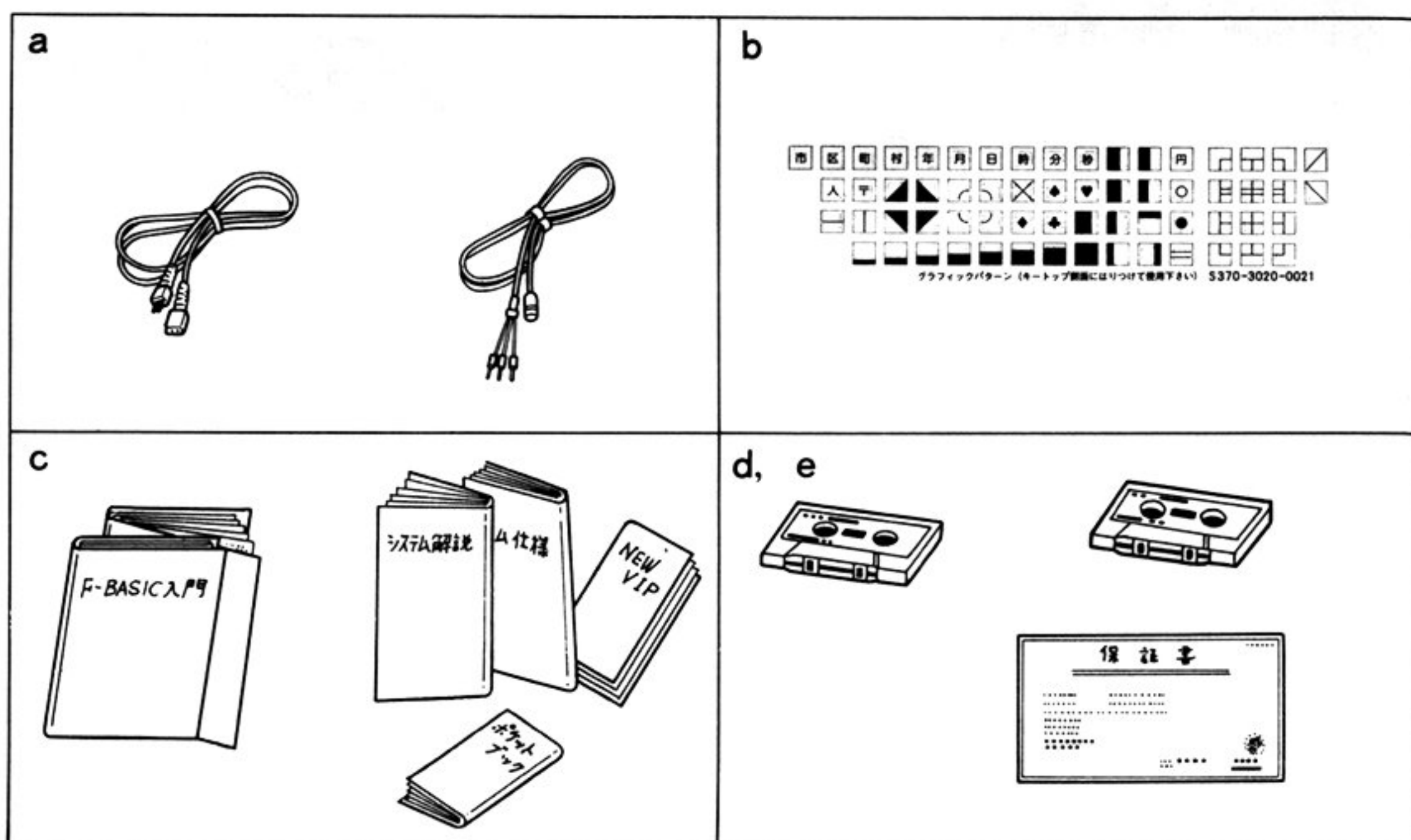
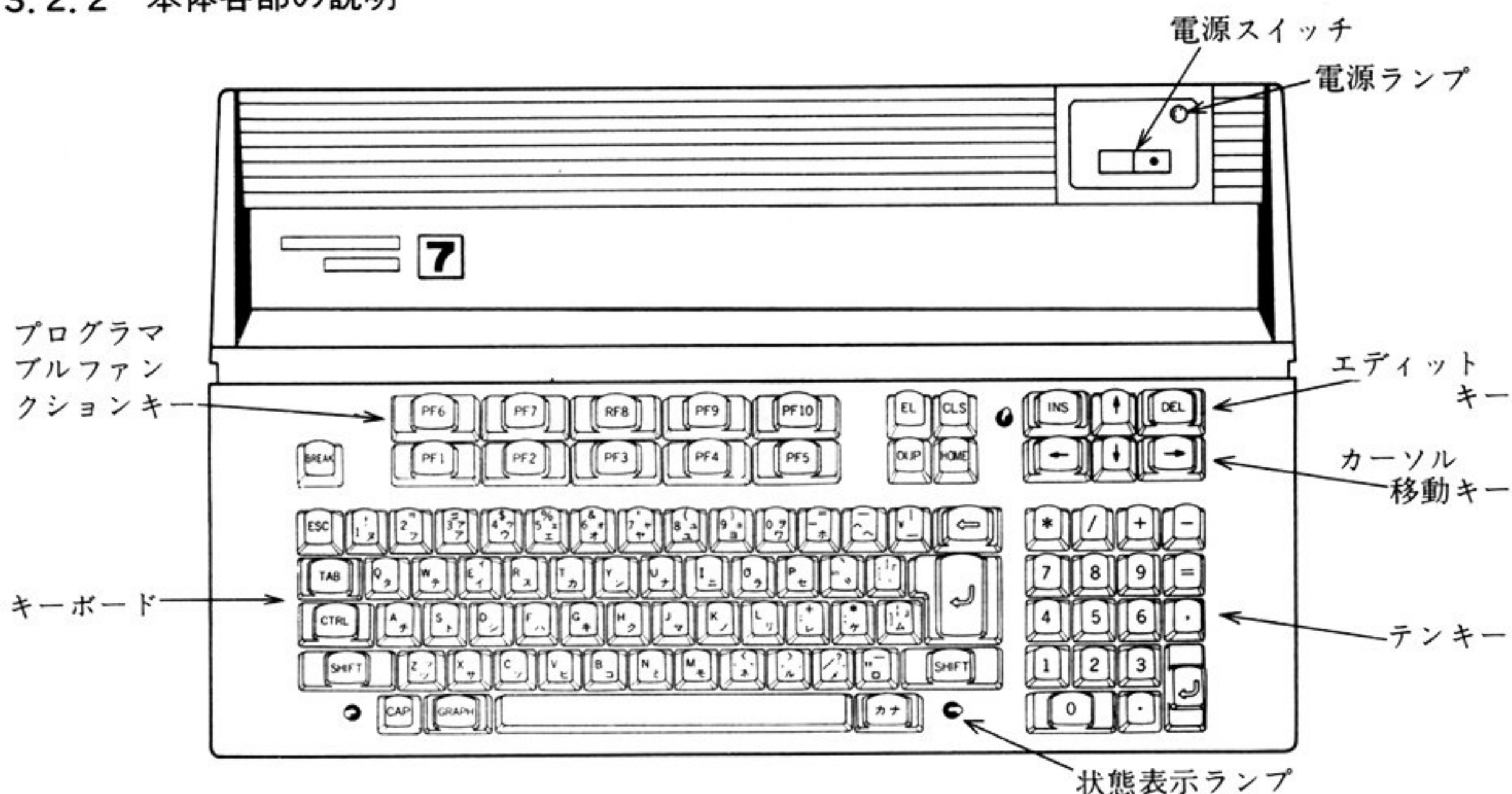
a

b

c

d，e

3.2.2 本体各部の説明

電源スイッチ
電源ランプ
プログラマブルファンクションキー
エディットキー
カーソル移動キー
キーボード
テンキー
状態表示ランプ

電源スイッチ

このスイッチを ON 側にたおしますと，本体の電源が入ります．スイッチの横には電源ランプがあり，電源が入りますとこのランプが点灯します．

キーボード

通常のタイプライタとほぼ同じ配列をしています．カナ文字の配列は，JISキーボードの配列に準拠しています．

英大・小文字，記号，カナ文字，句読点，グラフィック記号，数字がこのキーボードより入力できます．

テンキー

使用頻度の高い数字，演算記号を使いやすいようにまとめたキーです．キーボード文字列中にある数字や演算記号とはまったく同じ動作，作用をし，特別な区別はありません．

プログラマブルファンクションキー（PF キー）

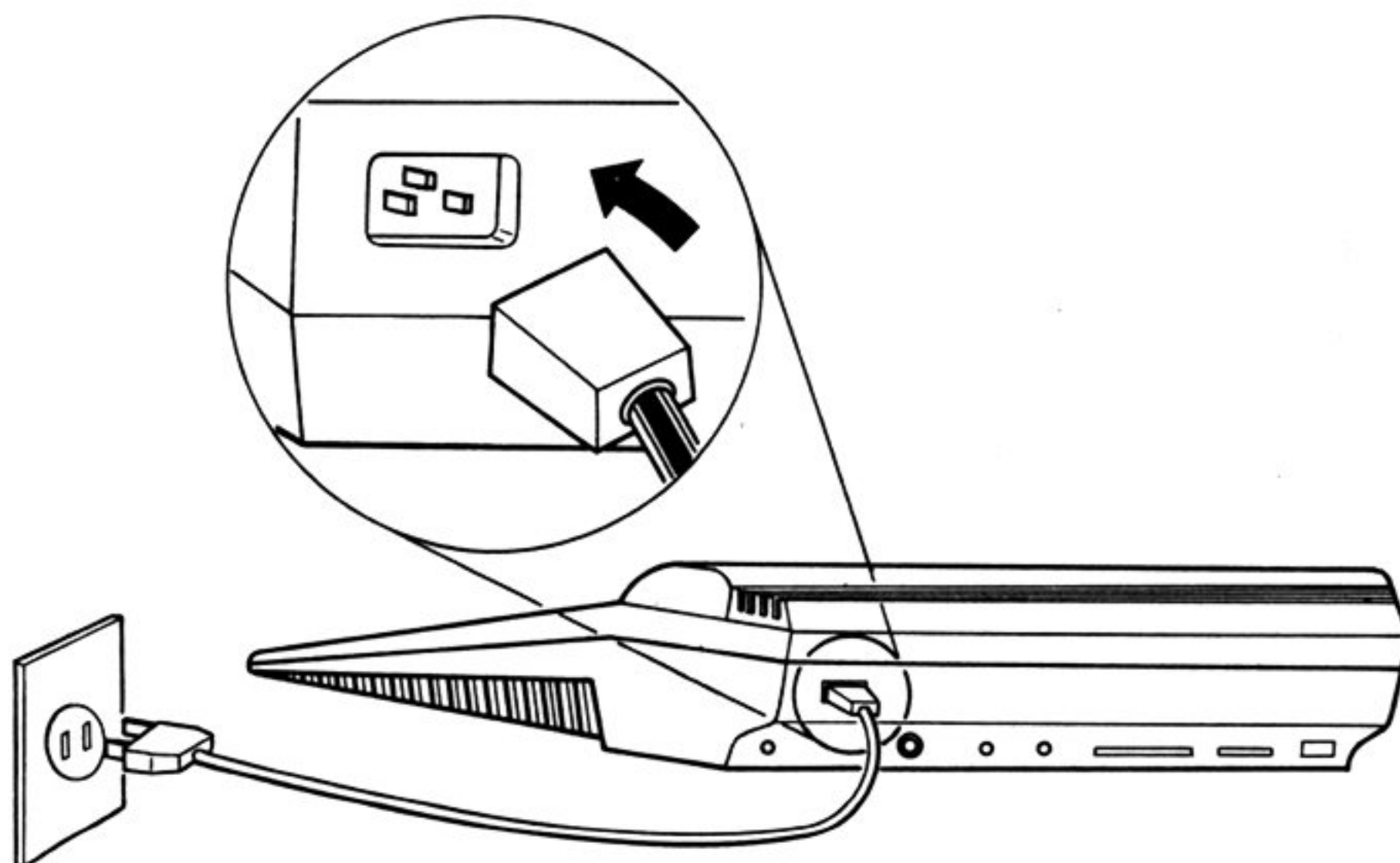
プログラム作成時や，アプリケーションプログラムの操作に使います．

エディットキー（INS・DEL 等）

プログラム作成，修正を行なうときに使用する機能キーです．

状態表示ランプ

INS の左横，CAP の左横 カナ の右横にあります．それぞれモードを持つキーで，ON・OFF の状態をランプで表示します．いずれもランプが点灯している状態が作動中(ON)です．

3.2.3 電源ケーブルの接続

電源ケーブルを本体背面の電源用コネクタに接続して下さい.

3.2.4 電源スイッチの投入

スイッチは, ON側へたおすとON状態となります. 電源スイッチがONにされますと電源ランプが点灯し, 本体に電源が投入されていることを示します.

周辺装置の電源をすべて投入した後, 本体の電源スイッチをONにして下さい. 電源の切断は, 本体の電源スイッチをOFFにした後, 周辺装置の電源をOFFにして下さい.

3.2.5 本体の背面

リセット スイッチ
このスイッチを押すと，リセットが行なわれ，RAM のプログラムエリアをクリアします．いったん電源を切ったのと同じ状態になります．
オーディオカセット コネクタ
オーディオカセットを使う時は，このコネクタとオーディオカセットを本体付属品のオーディオカセットケーブルで接続します．
カラー CRT コネクタ
カラー CRT を使う時は，このコネクタとカラー CRT とを専用ケーブルで接続します．
グリーン CRT コネクタ
グリーン CRT を使う時は，このコネクタとグリーン CRT とを専用ケーブルで接続します．
外部スピーカ用コネクタ
このコネクタに，市販のスピーカを接続します．
サウンド・ボリューム (RV1)
ブザー・ボリューム (RV2)
ブザー音，サウンド音の音量調整をするボリュームです．
I/O 拡張バスコネクタ
I/O 拡張ユニットとの接続用コネクタです．
プリンタ コネクタ
プリンタを使う時は，このコネクタとプリンタを専用ケーブルで接続します．
ディップ スイッチ
ブートローダの設定，クロック周波数の設定を行なうスイッチです．

3.3 CRTディスプレイ

<FM-7に使用可能なCRT>

品　　名	型　　番	備　　考
高解像度カラーCRTディスプレイ	MB27301	カラーCRTケーブル（MB26512）が必要.
カラーCRTディスプレイ	MB27303	カラーCRTケーブル（MB26512）が必要.
カラーCRTディスプレイ（TVグレード）	MB27305	カラーCRTケーブル（MB26512）が必要.
高解像度グリーンCRTディスプレイ	MB27302	グリーンCRTケーブル（MB26503）が必要.
グリーンCRTディスプレイ	MB27304	グリーンCRTケーブル（MB26503）が必要.
家庭用カラーテレビ（白黒も接続可能）	——	家庭用カラーテレビアダプタ（MB22602）が必要.

接続作業を行なう場合は，必ず各機器の電源をOFFの状態にして下さい．忘れずに確認を行なって下さい．

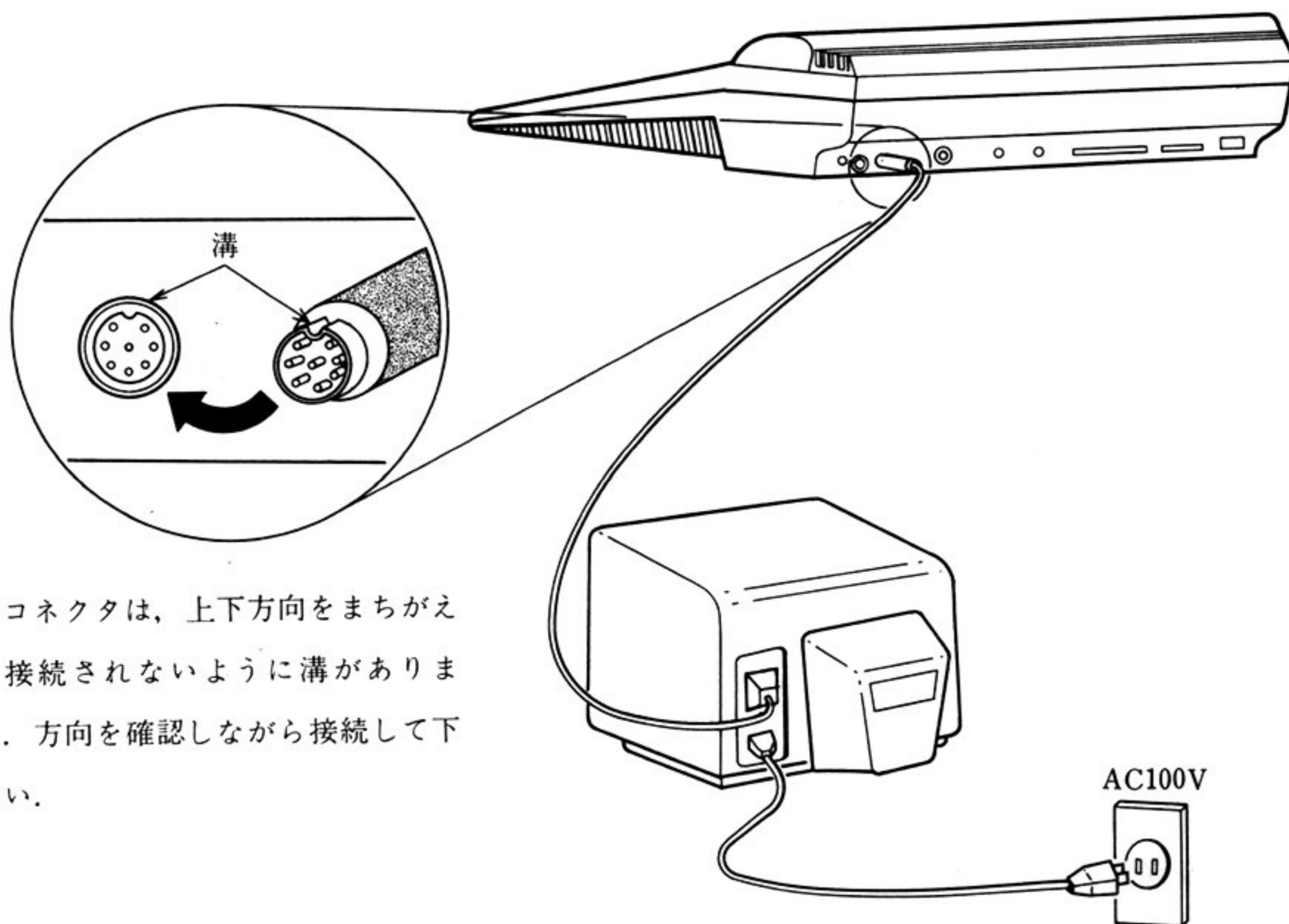
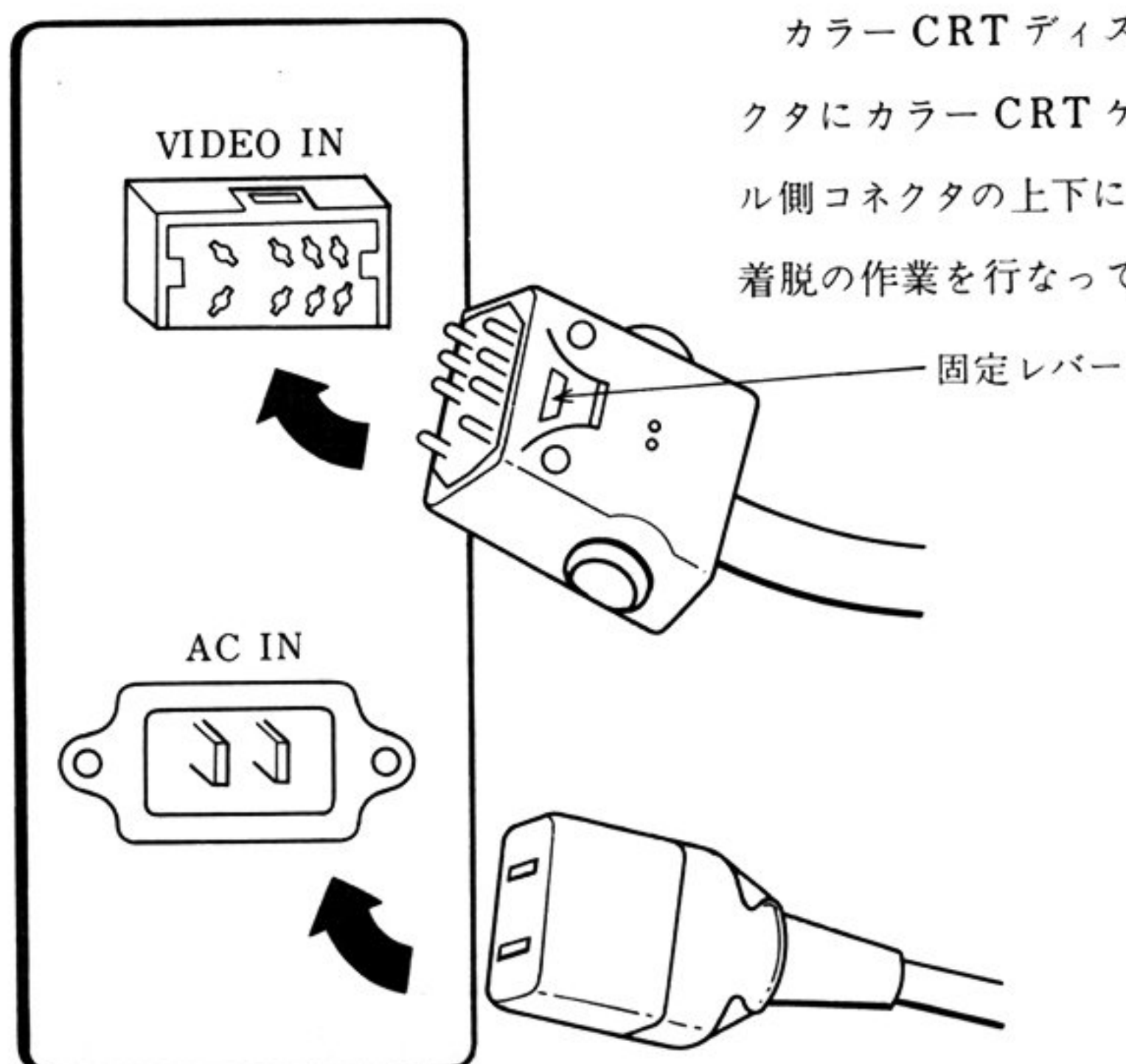
専用ディスプレイ以外のディスプレイ（RGB方式家庭用テレビも含む）を使用する際には，CRTインタフェースを充分に理解するか，または購入された販売店に御相談下さい．

（ただし，専用ディスプレイ以外のディスプレイ接続時の障害についての責任は負いかねます.）

3.3.1 カラーCRT ディスプレイの接続

カラーCRT と本体の接続は，カラーCRT ケーブルにて接続します．

溝

AC100V

コネクタは，上下方向をまちがえて接続されないように溝があります．方向を確認しながら接続して下さい．

カラーCRT ディスプレイ背面の，VIDEO IN コネクタにカラーCRT ケーブルを接続する際には，ケーブル側コネクタの上下にある，固定レバーを押した状態で，着脱の作業を行なって下さい．

VIDEO IN

固定レバー

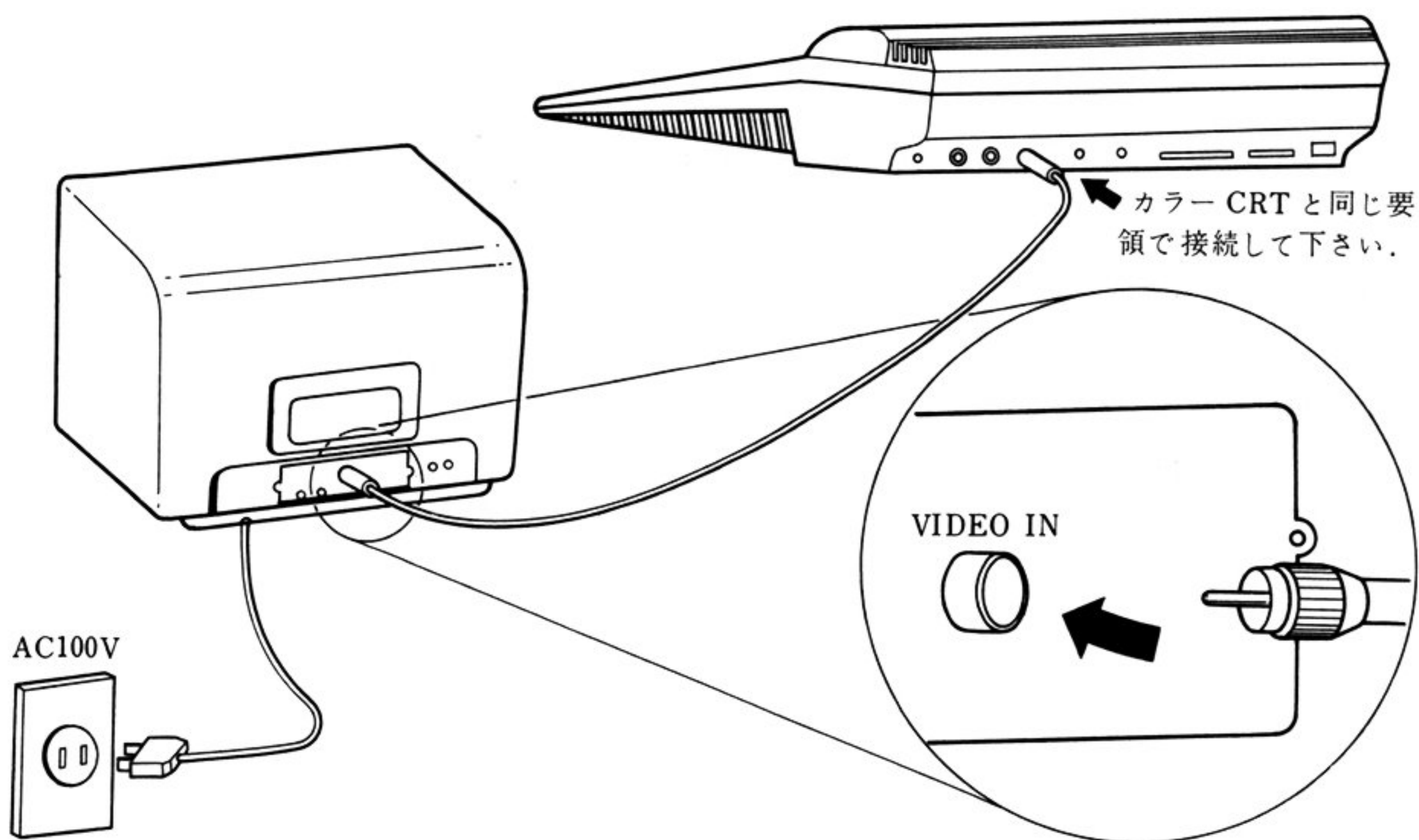
AC IN

3.3.2 グリーンCRT ディスプレイの接続

グリーン CRT と本体の接続はグリーン CRT ケーブルにて接続します.

カラー CRT と同じ要領で接続して下さい.

VIDEO IN

AC100V

3.3.3 家庭用テレビの接続

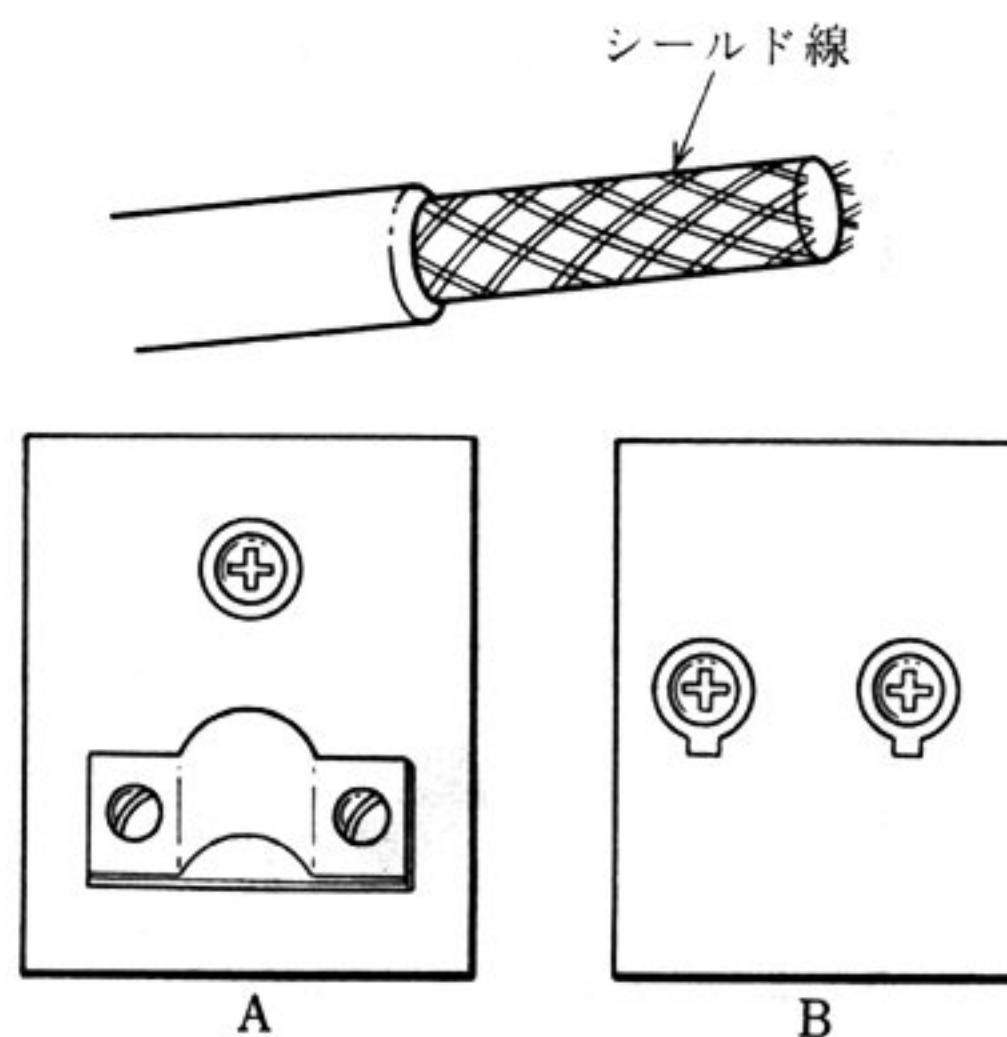
家庭用カラーテレビを，ディスプレイ装置として御使用される場合には，家庭用テレビアダプタ（オプション）が必要となります.

(1) 接　続

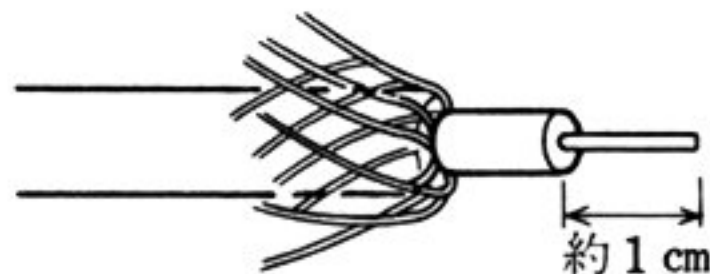
① テレビ接続ケーブルの端を家庭用テレビのアンテナ端子に接続します.

シールド線

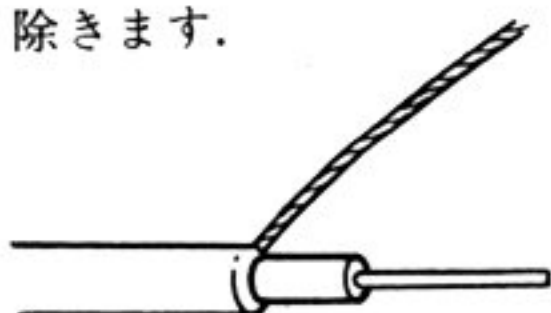
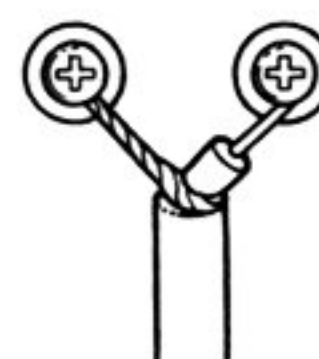
- ナイフなどで外側のビニールを 2 cm 程度取り除きます．すると網のようなシールド線が現われます.
- これより次の作業は，テレビの機種により異なります．テレビのアンテナ端子を見て下さい．アンテナ端子には大きく右の2種類があります．(VHF 用)

A　B

〈アンテナ端子がAの場合〉

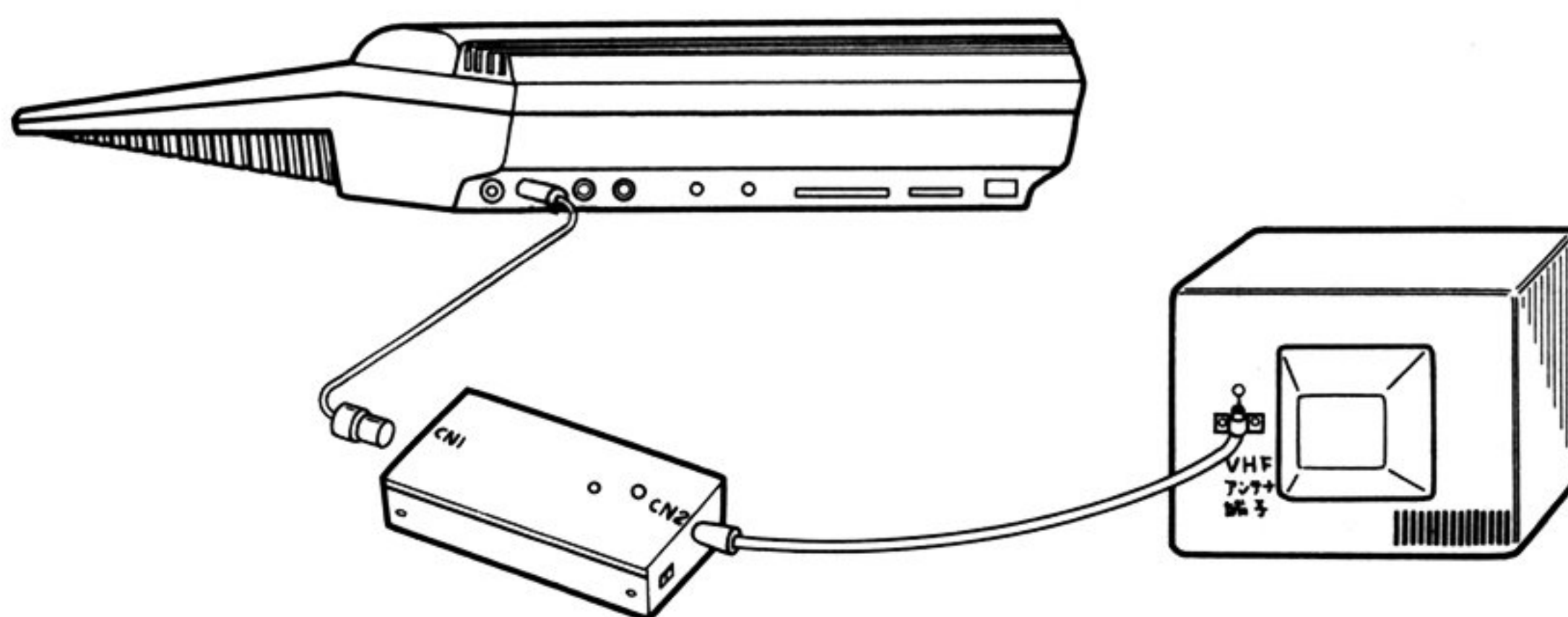
シールド線を，切断しないように，ピンセットなどでゆっくりほぐしてから，内側に折ります．次に銅線の被覆をナイフなどで1 cm程度取り除きます．

約1 cm

下図のようにVHFアンテナ端子に接続します．

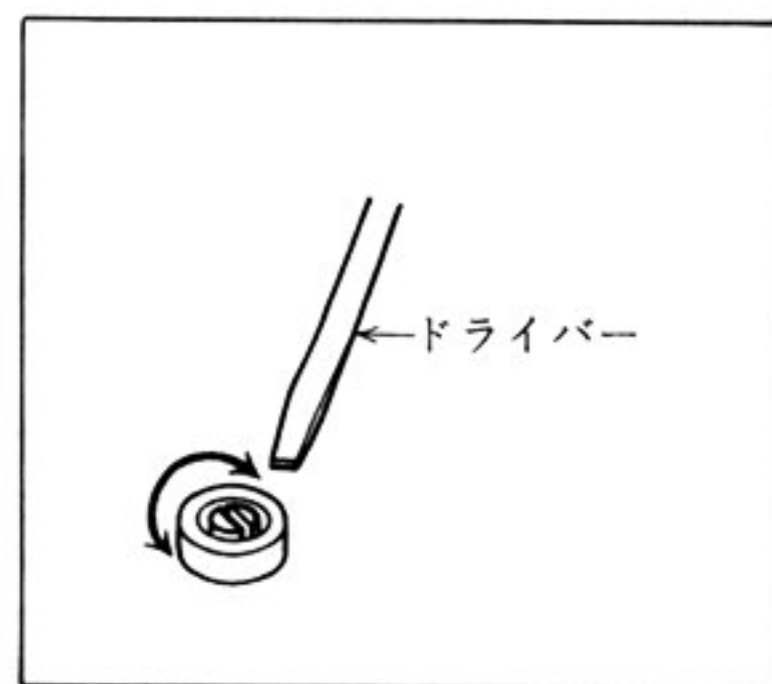
〈アンテナ端子がBの場合〉

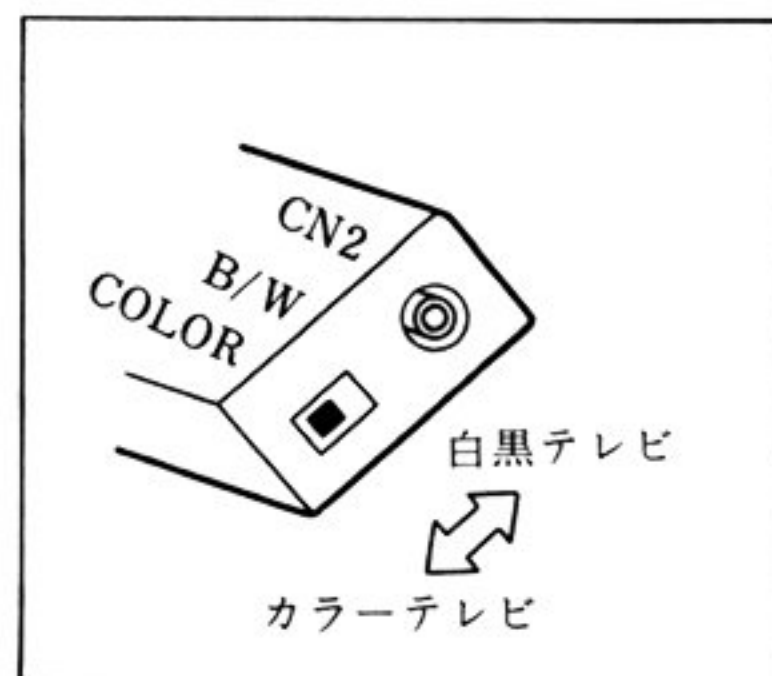
シールド線の網をピンセットなどではぐし，片方にまとめます．次に銅線の被覆をナイフなどで，1 cm程度取り除きます．

下図のようにVHFアンテナ端子に接続します．

● 注意事項 ●　作業は，シールド線や銅線を傷つけないよう，ていねいに行なって下さい．また，ナイフなどの扱いでけがなどされぬように充分御注意下さい．

② テレビ接続用ケーブルのコネクタをCN 2（コネクタ2）に接続し，CN 1には家庭用カラーテレビアダプタ用ケーブルの一方を，もう一方を，本体のカラーCRTのコネクタと接続して下さい．

CN1
CN2
VHF
アンテナ
端子

（2） 調　整

① カラー画質調整

画面に縞模様が現われ，鮮明さを欠くような場合には，「C.ADJ」と記してある，画質調整用コンデンサを，マイナスドライバーで左右に回し調整して下さい．

② チャンネル調整

テレビアダプタの出力は出荷時には，２チャンネルに合わせてあります．１チャンネルで使用になる場合には，販売店に申し付けて下さい.（調整時には専用ドライバーが必要です．）

③ カラー，白黒の選択方法

カラーテレビと接続した場合は，テレビアダプタのスイッチを「COLOR」にします．「B/W」にしますと，白黒の画面になります．白黒テレビの場合はどちら側でもかまいません．

ドライバー

CN2
B/W
COLOR
白黒テレビ
カラーテレビ

（3） 使用上の注意

① アンテナの共同受信を御利用になっている場合には，アンテナ線をおはずしの上御使用下さい．共同受信を受けている他の画像が乱れる恐れがあります．

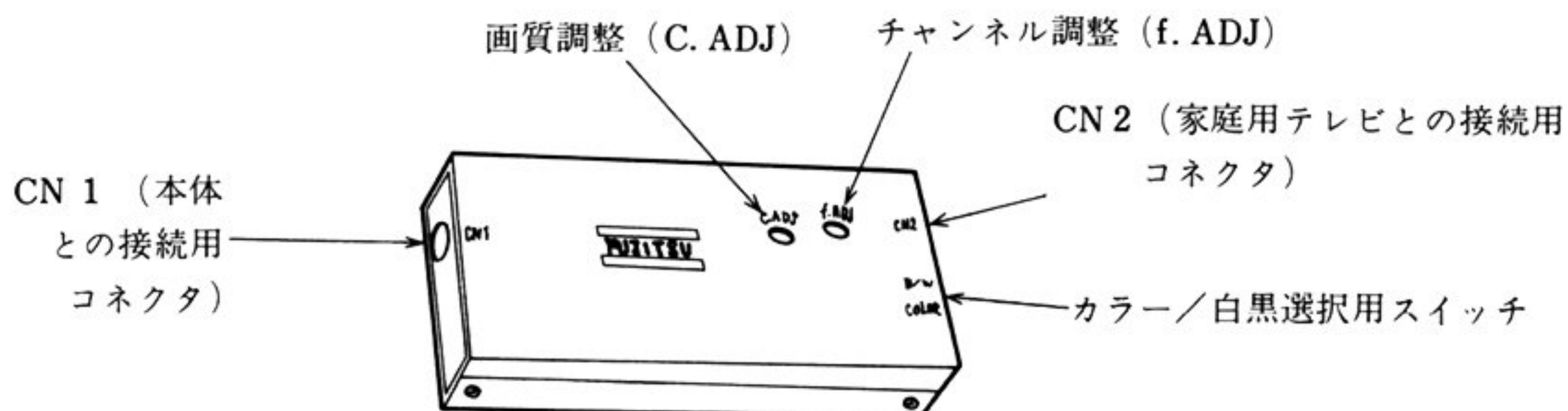
② 本製品は精密部品が組み込まれていますので，接続調整などの取扱いはていねいに行なって下さい．

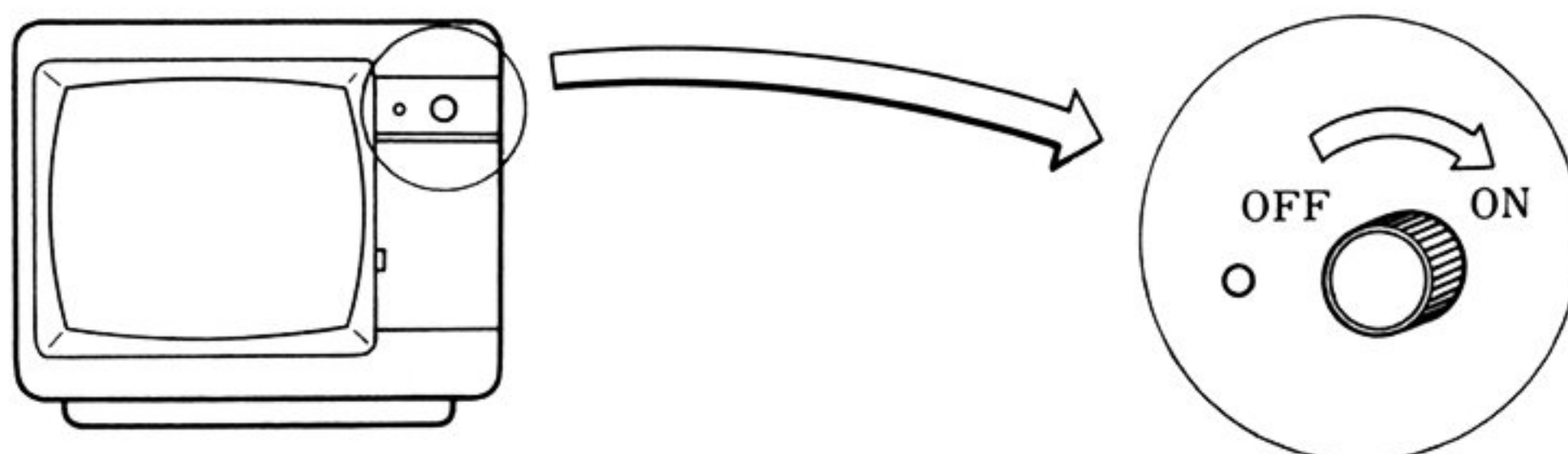
③ テレビアンテナ端子は，VHF アンテナ端子を御使用になって下さい．UHF アンテナ端子では御使用になれません．

④ 「CN 2」の端子コネクタは必要以外には（やむをえない機器の移動など）着脱を行なわないで下さい．接触不良の原因になりかねません．

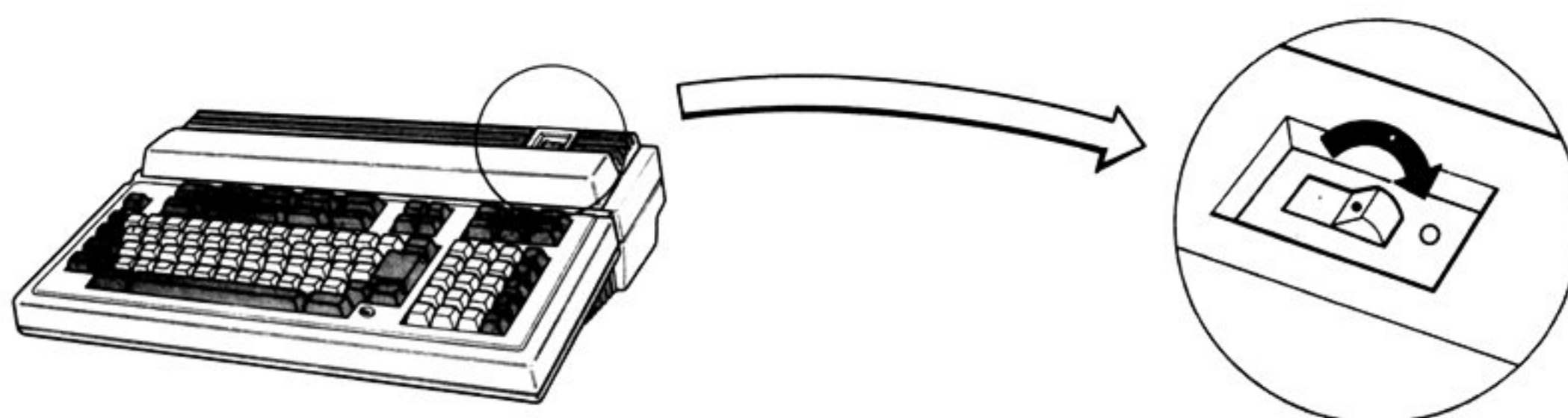
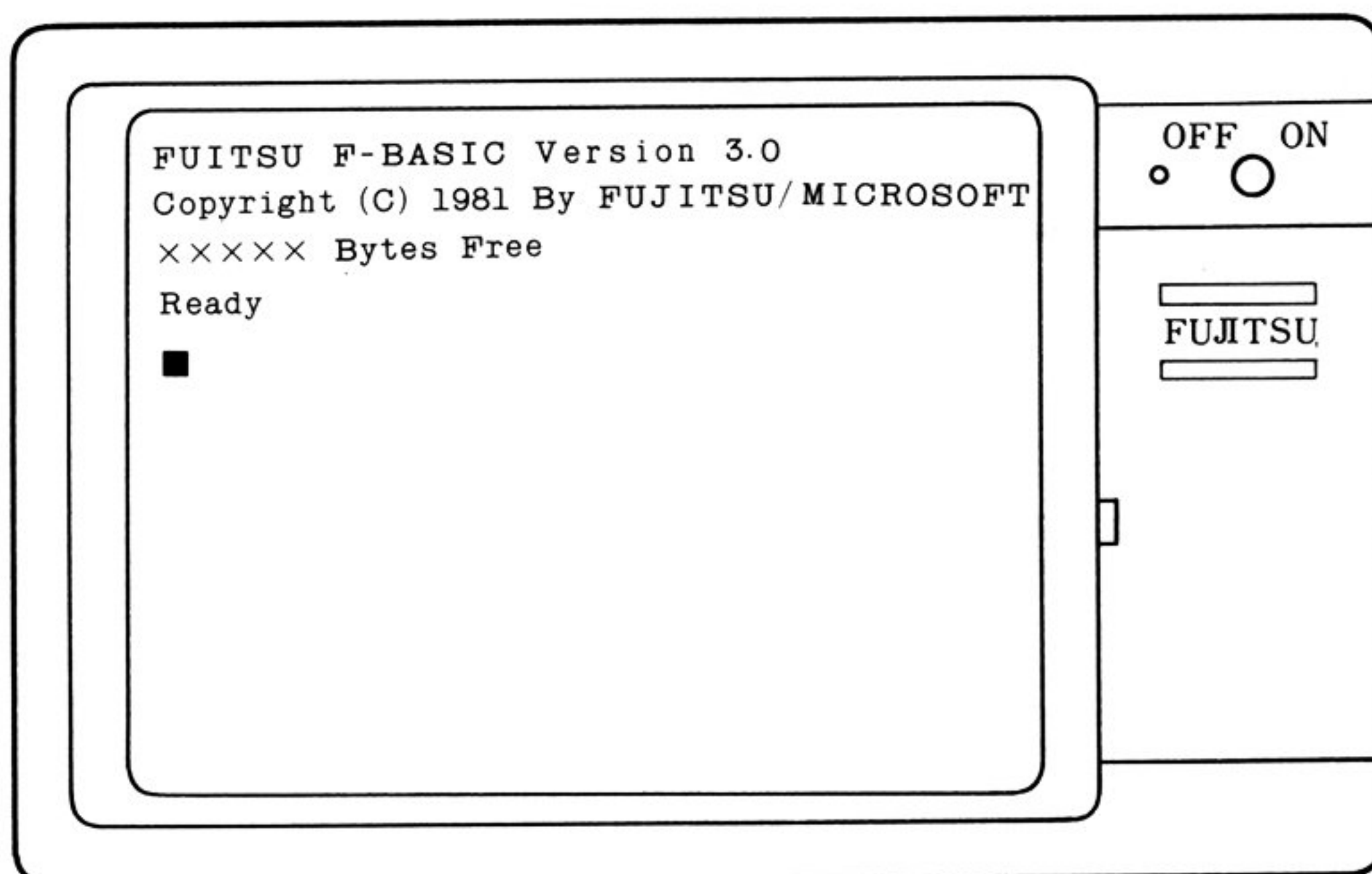
⑤ チャンネル調整の「f.ADJ」は，専用ドライバー以外の物では調整できません．マイナスドライバーなどで調整をしますと，コアが破損する恐れがあります．チャンネル変更をなさるときは，必ず販売店にお申し付け下さい．

⑥ フタは閉めて御使用下さい．

画質調整（C. ADJ）
チャンネル調整（f. ADJ）
CN 2（家庭用テレビとの接続用コネクタ）
CN 1（本体との接続用コネクタ）
カラー／白黒選択用スイッチ

3.3.4 CRT ディスプレイのテスト

(1) CRTディスプレイの電源スイッチを入れて下さい．電源スイッチを右側ONの方向につまみを回して下さい．電源が入りますと，電源表示ランプが点灯し，電源が入ったことが確認できます．

(2) CRTディスプレイの電源が入っていることを確かめてから，次に本体の電源を入れます．

(3) 接続作業が正しく行なわれていますと，次の図のような表示が表われます．

```
FUITSU F-BASIC Version 3.0
Copyright (C) 1981 By FUJITSU/MICROSOFT
××××× Bytes Free
Ready
■
```

〈写らなかった場合の対処方法〉

（1） 表示が薄く見づらい

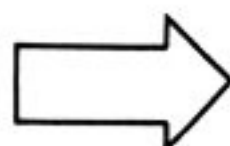
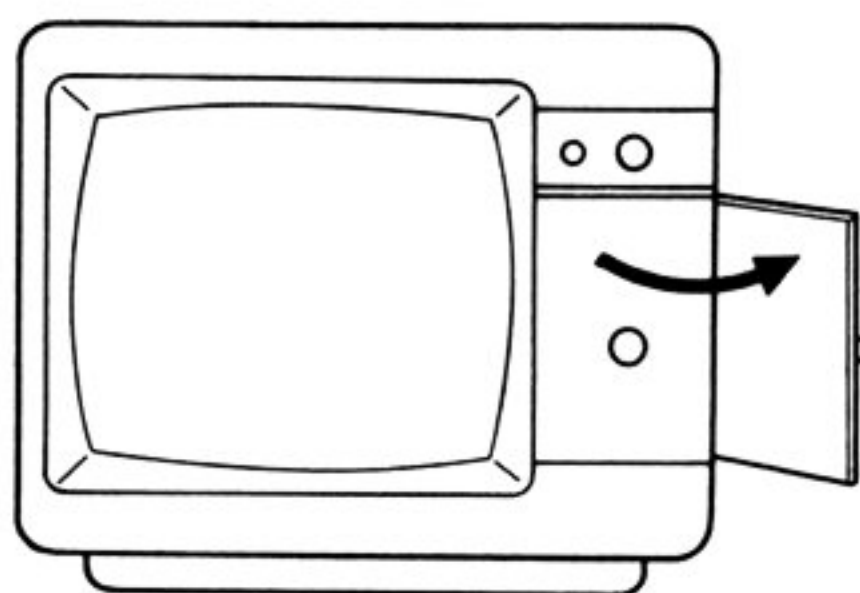
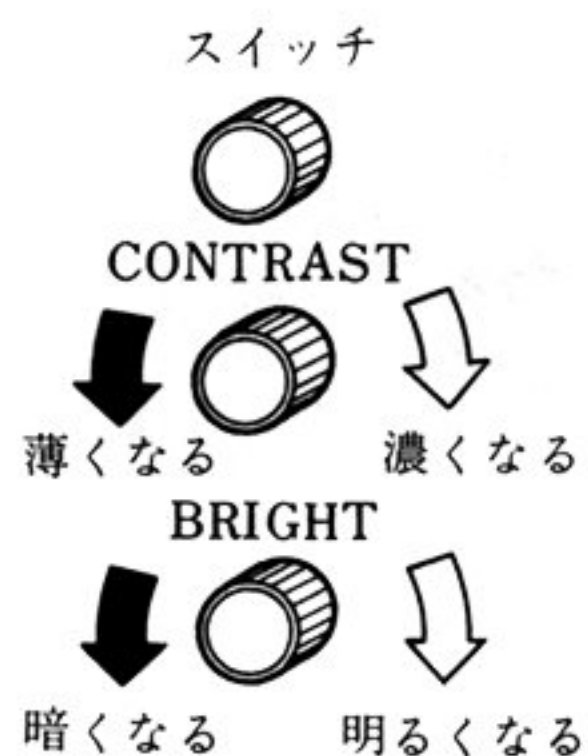
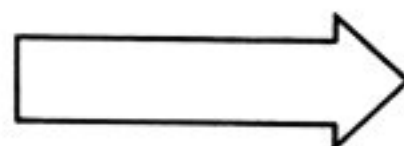
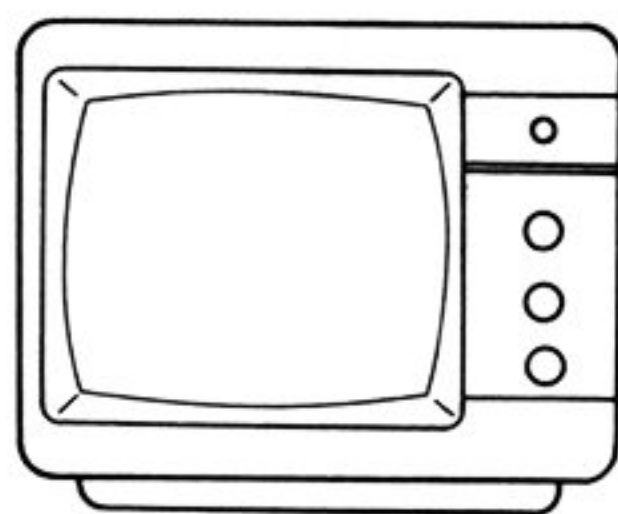
対処 ブライト（BRIGHT），コントラスト（CONTRAST）の修正を行なって下さい．専用ディスプレイの調整つまみは，左右にそれ以上回らない所が最大です．

輝度（BRIGHT）が強すぎると，目が疲れやすいので充分御注意下さい．

カラーCRTディスプレイにはコントラストはありません．

BRIGHT
暗くなる
明るくなる

カラーCRT ディスプレイ

スイッチ
CONTRAST
薄くなる
濃くなる
BRIGHT
暗くなる
明るくなる

グリーンCRT ディスプレイ

（2） 全く写らない

対処 ① 電源ランプは点灯していますか？（本体，CRTディスプレイ）

② ケーブルは接続されていますか？　また，まちがった所にケーブルが接続されていませんか？

（3） （1），（2）とも該当しない．（1），（2）とも確認したがやはり写らない．

対処 購入された販売店に御相談下さい．

3.4 プリンタ

<FM-7 に使用可能なプリンタ>

品名	型番	備考
ビジネスプリンタ	MB27402	プリンタケーブルは同梱.
フロントインサータ	MB26201	ビジネスプリンタとともに使用.
シリアルドットプリンタ	MB27403	プリンタケーブル（MB26514）が必要.
ドットプリンタ	MB27404	プリンタケーブル（MB26514）が必要.

接続作業を行なう場合は，必ず各機器の電源をOFFの状態にして下さい．忘れずに確認を行なって下さい．

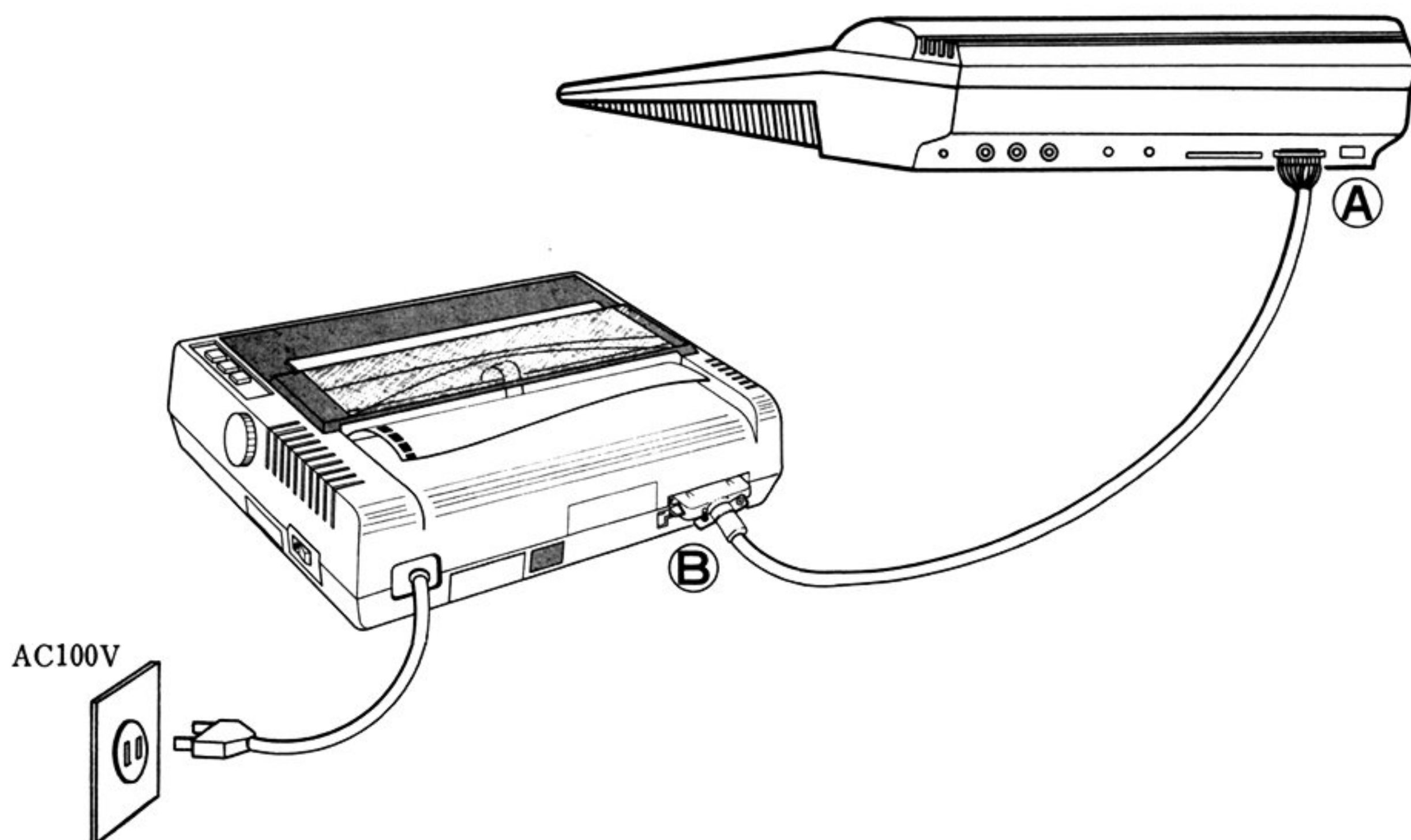
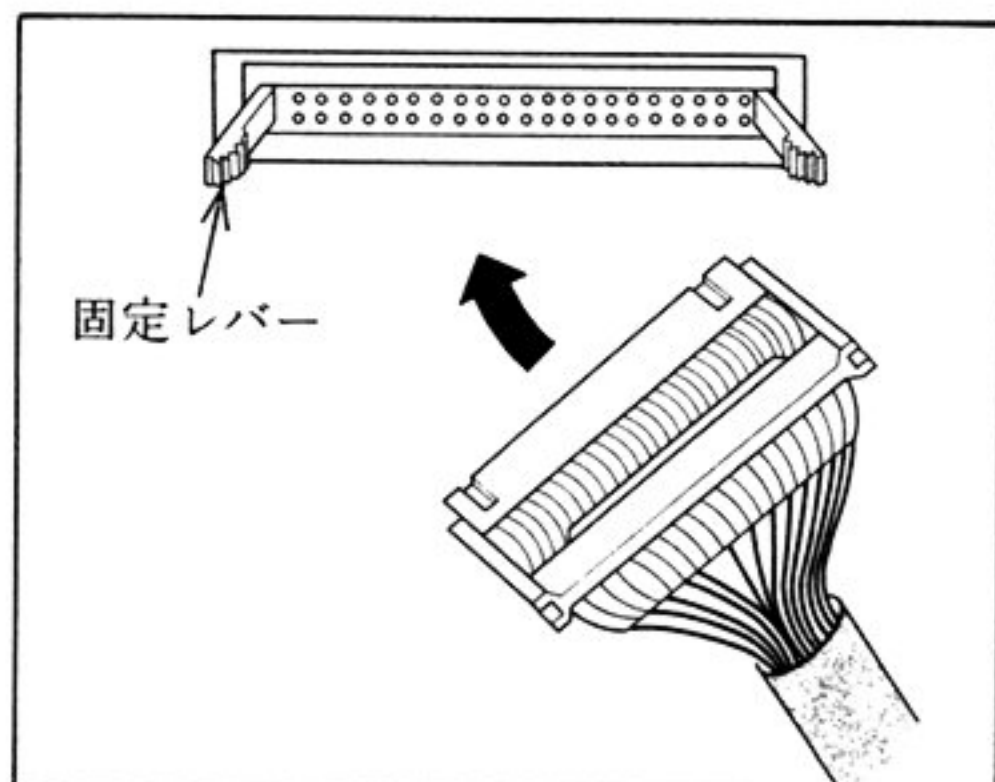
専用プリンタ以外のプリンタを使用する際には，プリンタインタフェースを充分に理解するか，または購入された販売店に御相談下さい．(ただし，専用プリンタ以外のプリンタ接続時の障害についての責任は負いかねます．)

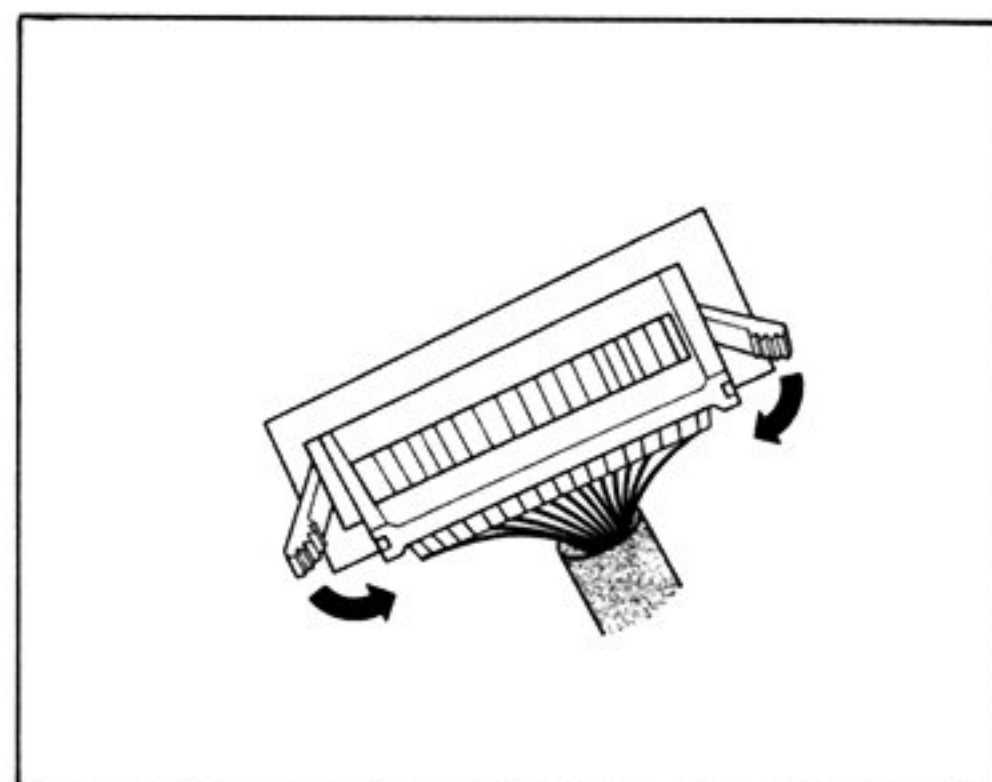
3.4.1　シリアルドットプリンタの接続

本体との接続はプリンタ用ケーブル (MB 26514) にて行ないます.

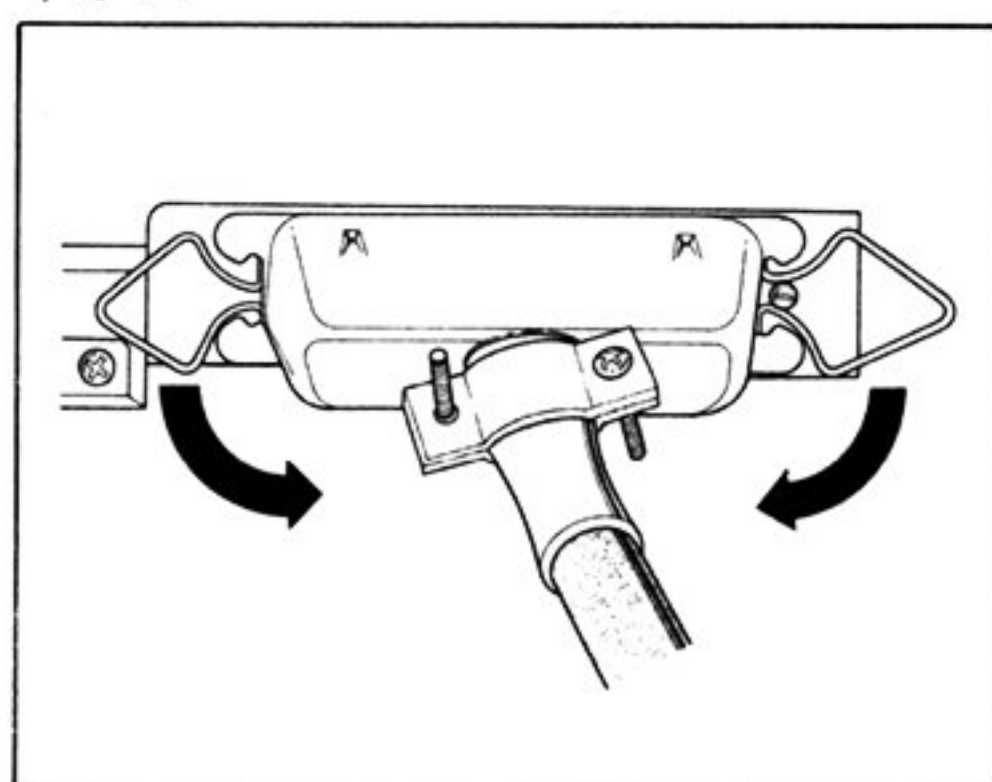
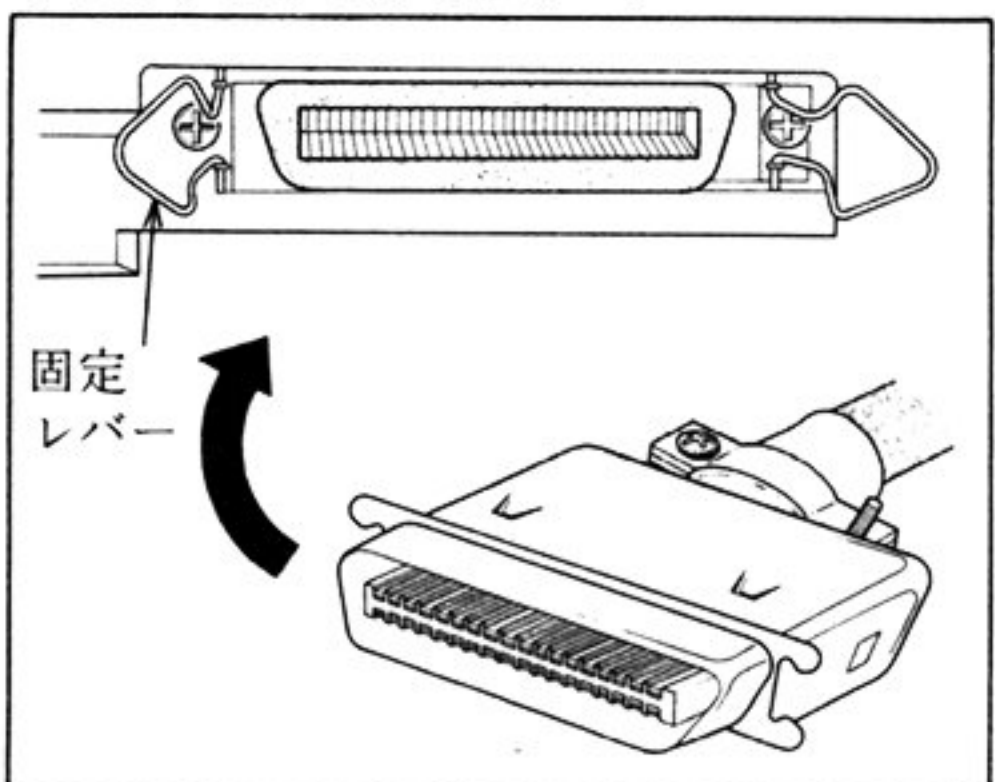
AC100V

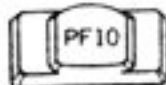
Ⓐ

Ⓑ

Ⓐ 本体との接続部

固定レバー

コネクタの方向をまちがえぬようにして，しっかりさしこんで下さい.

固定レバーを手前側に，カチッとはめて下さい.

Ⓑ プリンタとの接続部

固定レバー

3.4.2 シリアルドットプリンタのテスト

(1) テスト1(セルフテスト)

プリンタ自身の動作が正常であるかどうかを確かめるテストです．LFスイッチを押しながら電源スイッチを入れて下さい．セルフテストが開始され，次のようなテスト印字が行なわれます．印字は電源が切られるまで続けられます．

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmno
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnop
"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopq
#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqr
$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrs
%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrst
&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstu
'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuv
()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvw
)*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwx
*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxy
+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz
,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{
-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}
/0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~
0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_
123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~__
23456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~___
3456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~____
456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_____
56789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~______
6789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_______
789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~________
89:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_________
9:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_________|
:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_________||
;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_________|||
<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_________||||
=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[¥]^_`abcdefghijklmnopqrstuvwxyz{|}~_________|||||
```

チェックポイント

① プリンタヘッド部の動作に異常はないでしょうか？

② 正しく改行をしていますか？

③ 文字は正しく印字されていますか？(乱れていないか)

④ リボンは改行されるたびにリボンフィードされていますか？

(2) テスト2(接続テスト)

プリンタと本体が正しく接続されているかどうかを確かめるテストです．プリンタ，ディスプレイ，本体の順番で電源を入れ，Ready 状態になりましたら [PF10] のキーを押して下さい．正しく接続されていますと画面のハードコピーがプリントアウトされます．

* プリンタへの出力方法について詳しく知りたい場合は，第14章をお読み下さい．

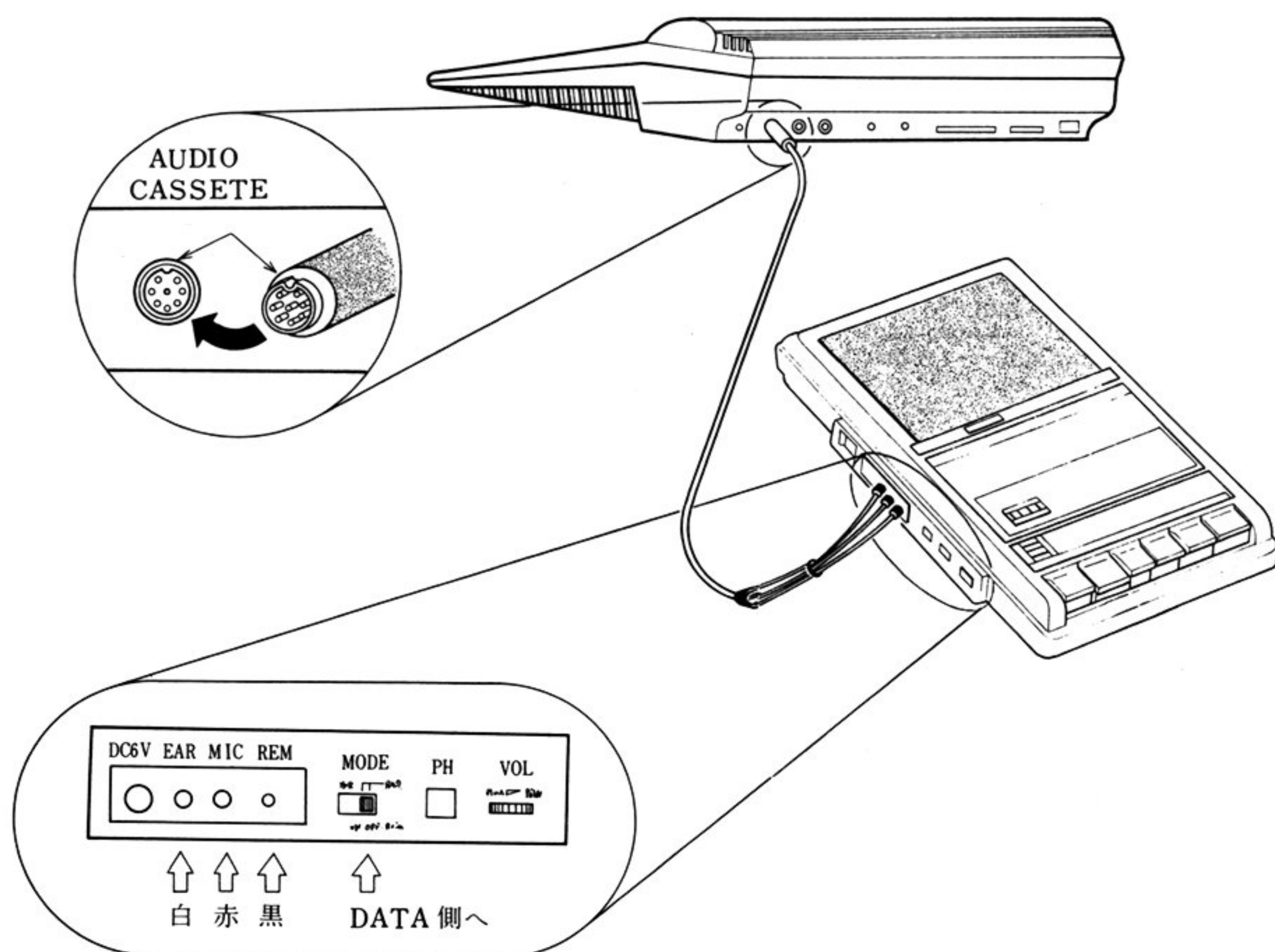
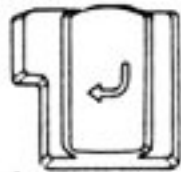
3.5 カセットテープレコーダ

3.5.1 FMデータレコーダの接続

FMデータレコーダ（MB27501）と本体との接続は，本体に添付されている専用ケーブル（MB26501）を使用して行ないます．

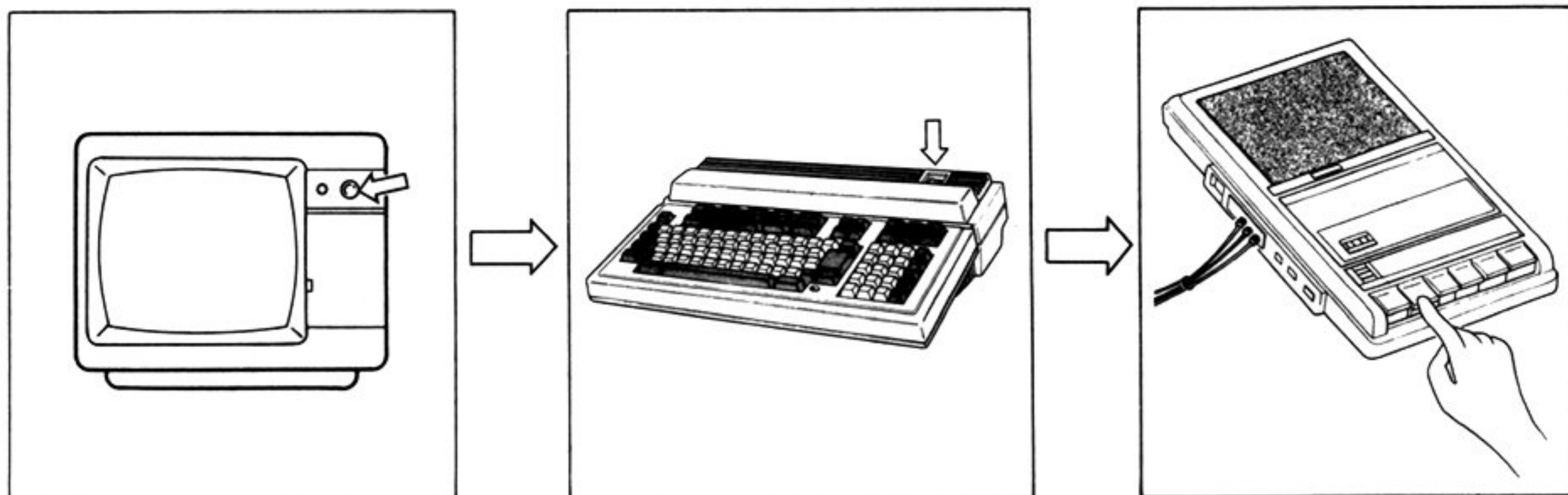
AUDIO
CASSETE

DC6V EAR MIC REM
MODE
PH
VOL

白 赤 黒　DATA側へ

3.5.2 FMデータレコーダのテスト

FMデータレコーダと本体が正しく接続されているかどうかのテストを行なって下さい.

① CRTディスプレイ，本体の順番で電源をONにして，DEMOテープをセットして下さい.

② Ready状態になっていることを確認してから，FMデータレコーダのPLAYスイッチを押して下さい．リモート機能が働いているので，テープは停止しています.（Ready状態：ディスプレイにReadyと表示され，カーソルが点滅している状態）

③ キーボードより次のように入力して下さい.

RUN "CAS0:DEMO1" ↵

↵キーを押すと同時に，FMデータレコーダのモータが回転し，カセットテープが走行します.

```
Ready
RUN "CAS0:DEMO1"
```

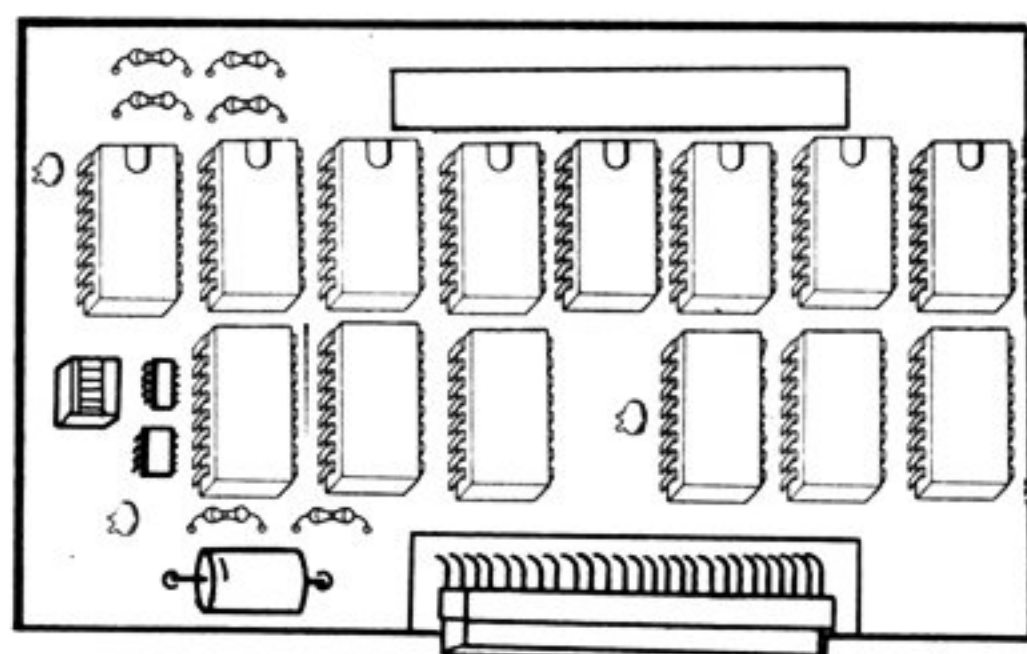
④ プログラムが捜し出されると，右のように表示され，LOADが完了するとプログラムが自動的に実行されます.

```
Ready
RUN "CAS0:DEMO1"
Searching
Found DEMO1
```

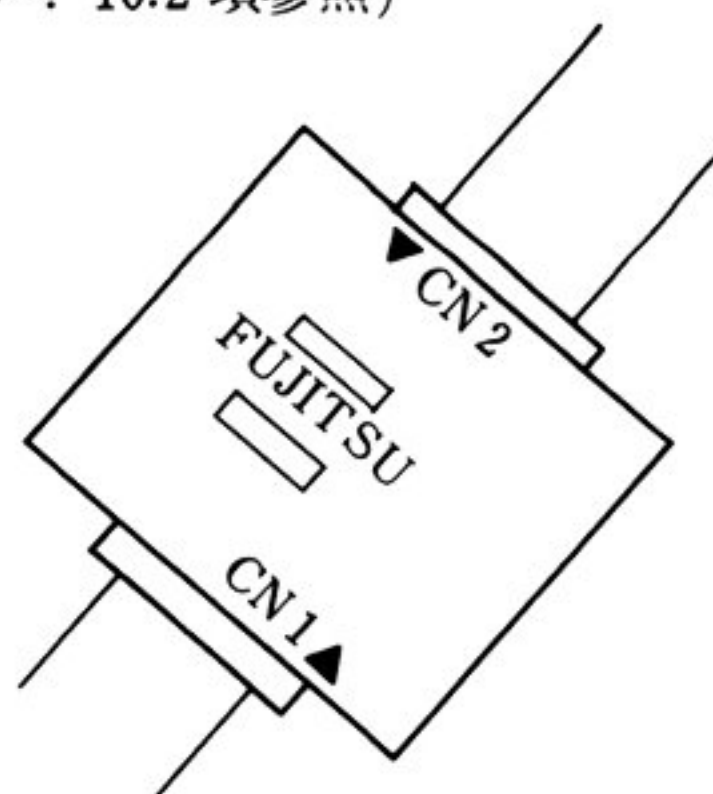
* カセットテープレコーダの扱い方については，第12章をお読み下さい.

3.6 ミニ フロッピィディスク ユニット

ミニフロッピィディスクユニットを本体に接続する場合には，ミニフロッピィインタフェースカードが必要です．ミニフロッピィインタフェースカード（以下インタフェースカードと略す）は，本体上部のトップカバーをはずし，インタフェースカードを装着します．

すでに，ミニフロッピィディスクユニットと，FM-8用に使われていたミニフロッピィアダプタをお持ちの方は，インタフェースカードをお求めにならずに，本体背面にあるI/O拡張コネクタに，ミニフロッピィアダプタを接続して御使用になれます．（ミニフロッピィアダプタにて使用の場合には，動作クロック周波数を，標準動作モードにてお使い下さい．10.2 項参照）

ミニフロッピィインタフェースカード

ミニフロッピィアダプタ

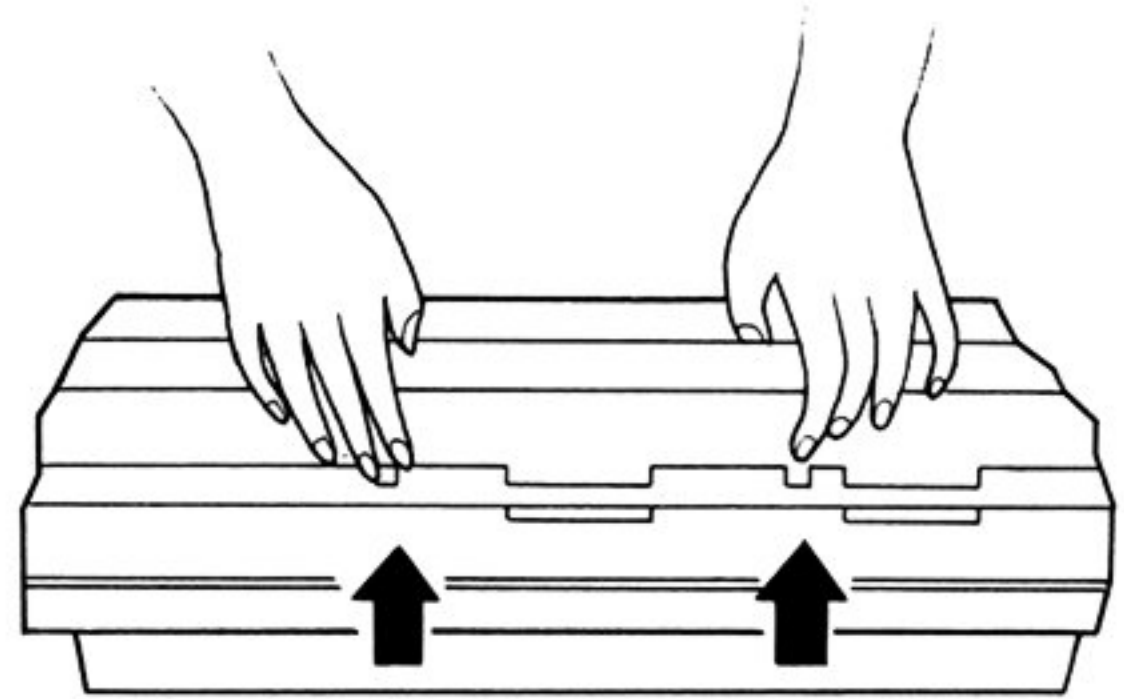
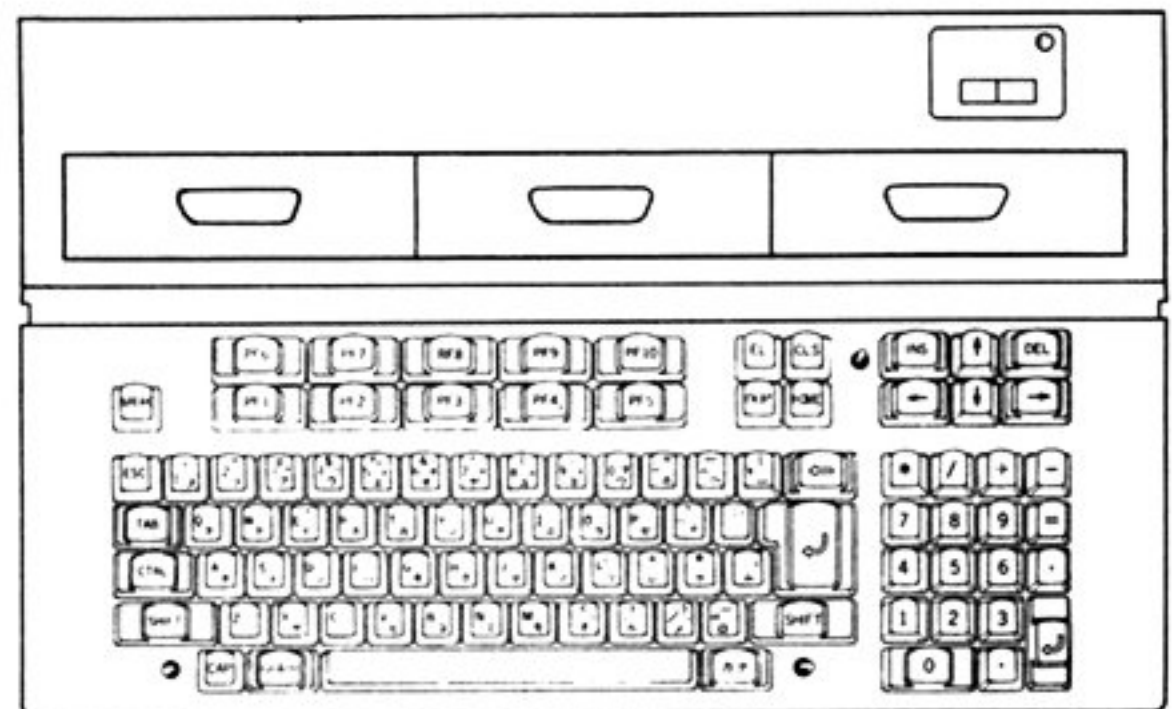
<FM-7に使用可能なミニフロッピィディスクユニット>

品　名	型　番	備　考
ミニフロッピィディスクユニット	MB27601	ミニフロッピィアダプタ(MB22603)，またはミニフロッピィインタフェースカード(MB22407)が必要.
Sタイプミニ FD ユニット	MB27605	ミニフロッピィアダプタ(MB22603)，またはミニフロッピィインタフェースカード(MB22407)が必要.
薄形ミニ FD ユニット	MB27607	ミニフロッピィアダプタ(MB22603)，またはミニフロッピィインタフェースカード(MB22407)が必要.
増設用薄形ミニ FD ユニット	MB27608	薄形ミニ FD ユニット(MB27607)の増設用.

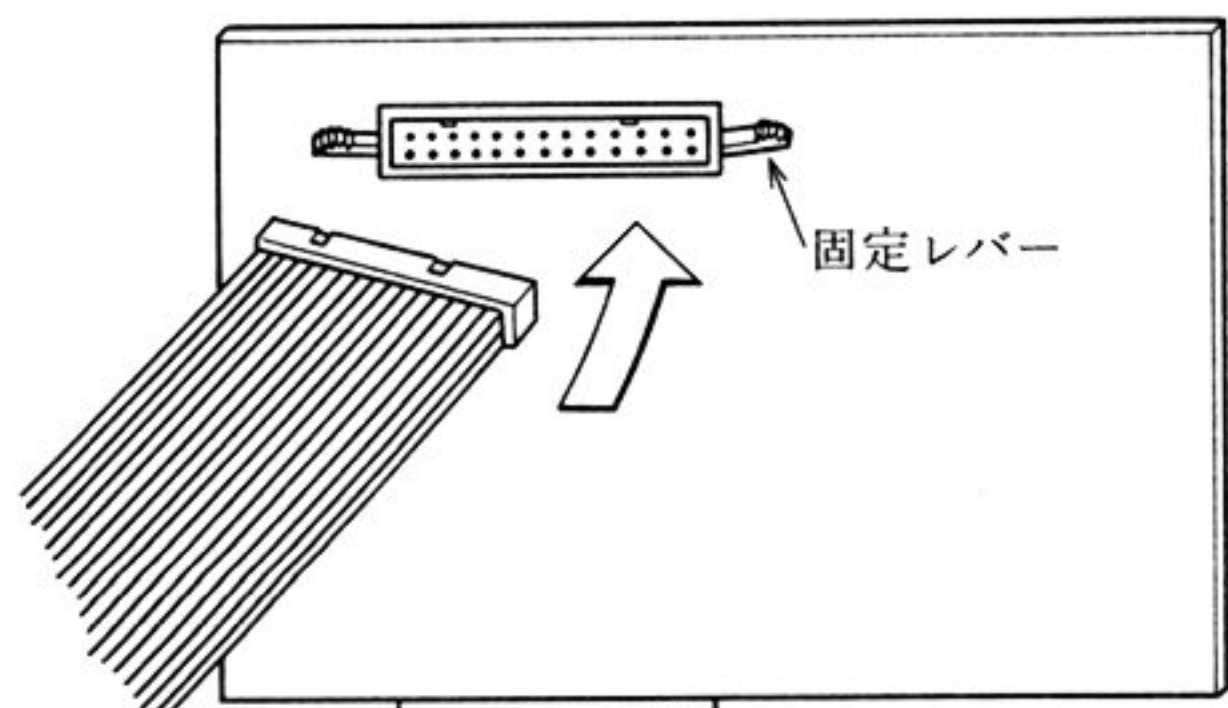
接続作業を行なう場合は，必ず各機器の電源をOFFの状態にして下さい．忘れずに確認を行なって下さい．

専用ミニフロッピィディスクユニット以外のミニフロッピィディスクユニットを使用する際には，ミニフロッピィディスクインタフェースを充分に理解するか，または購入された販売店に御相談下さい．(ただし，専用ミニフロッピィディスクユニット以外のミニフロッピィディスクユニット接続時の障害についての責任は負いかねます．)

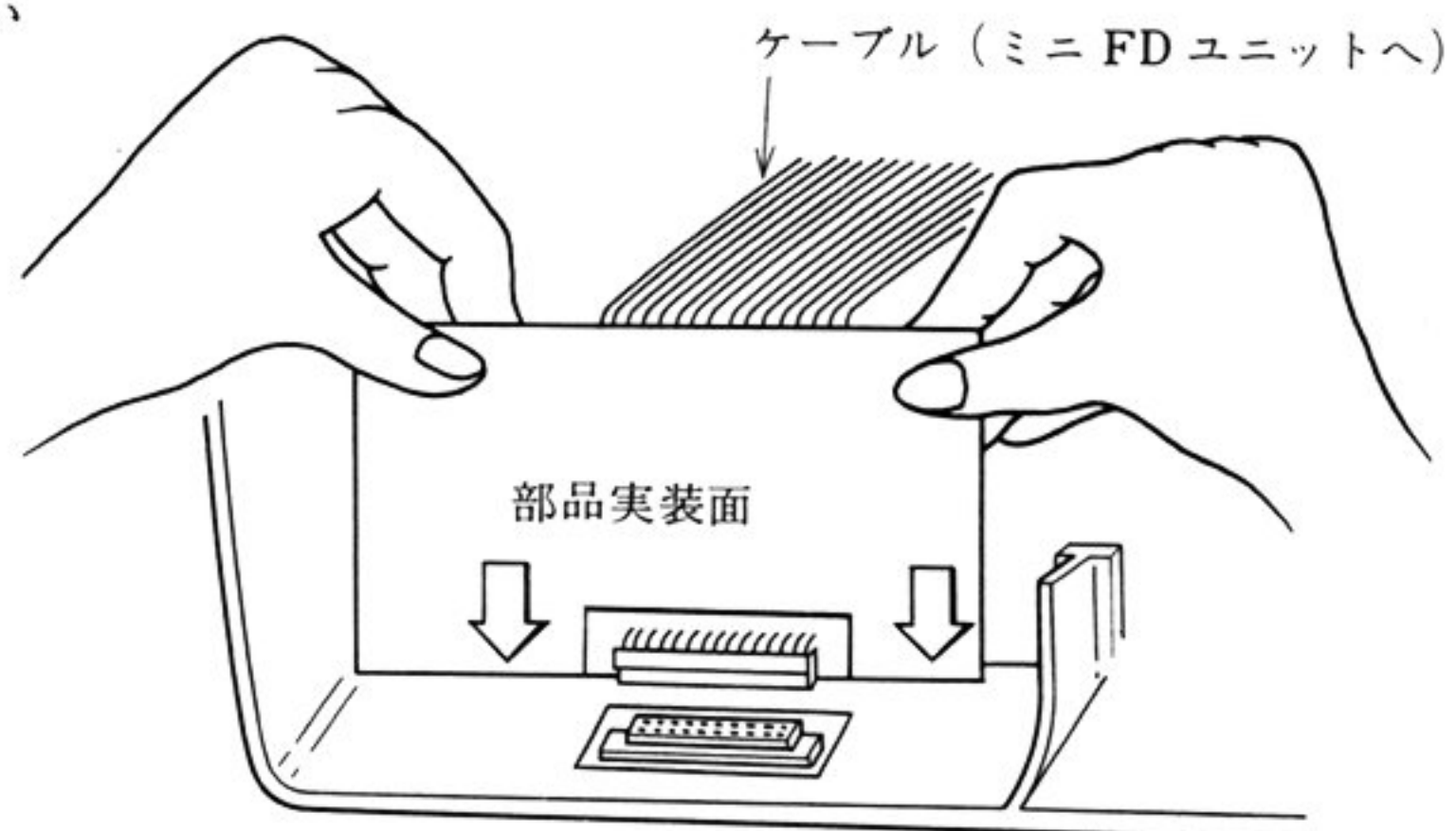
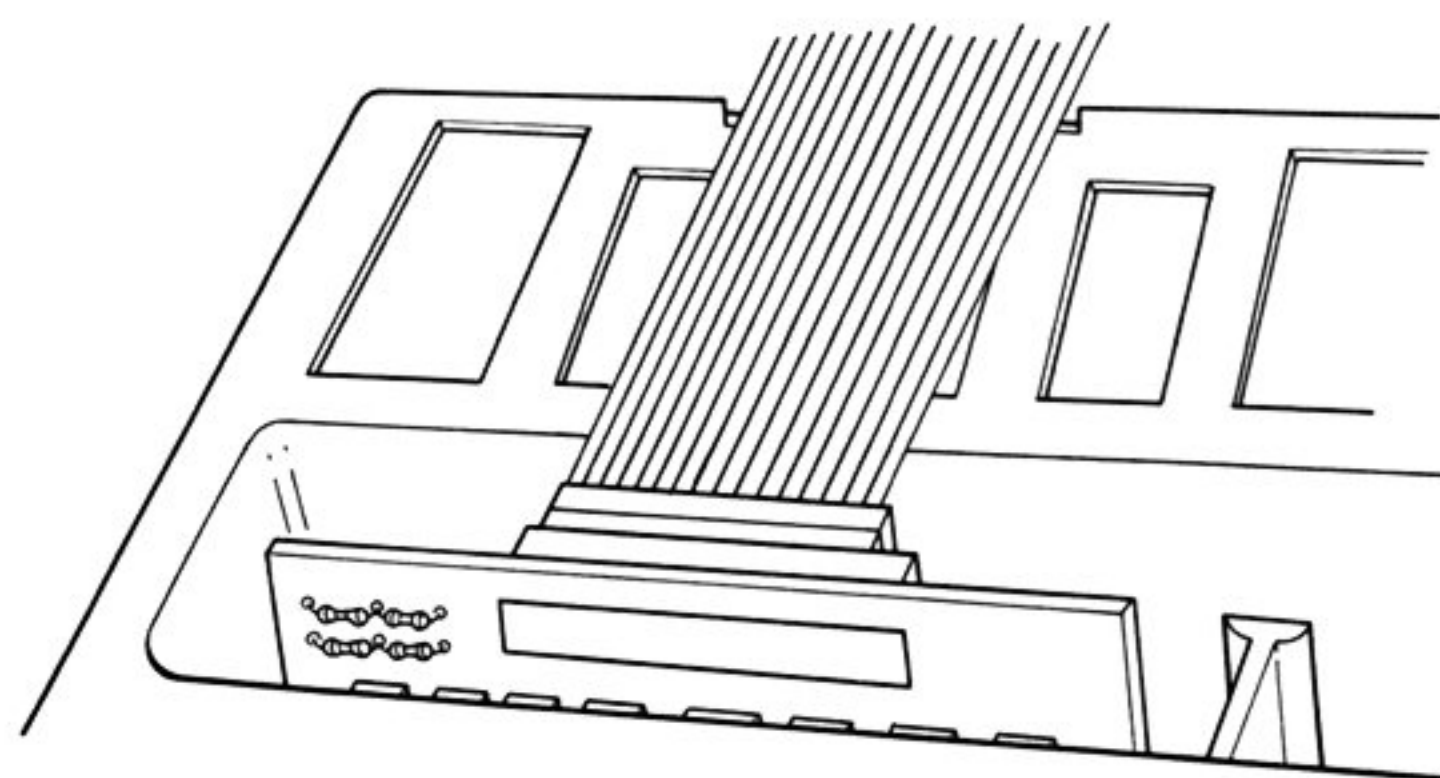
3.6.1 ミニフロッピィインタフェースカードの装着

本体上部のトップカバーをはずしますと，底にコネクタが3つ並んでいます．左側，中央のいずれかに（どちらでもかまわない）ミニフロッピィインタフェースカードを装着します．

① カバー後部にある固定爪を押しながら，ゆっくりカバーをはずします．固定爪の取扱いはていねいに行なって下さい．

② カバーを開けますと，右のように，3個コネクタが並んでいます．左側，中央のいずれかにミニフロッピィインタフェースカードをセットします．

③ ミニフロッピィインタフェースカード背面にあるケーブル用コネクタに，フロッピィケーブルをセットします．コネクタには逆差し防止溝があります．方向をよく確かめてセットして下さい．

固定レバー

④ 本体にインタフェースカードを装着します．部品実装面を手前側にし，コネクタに注意しながら接続して下さい

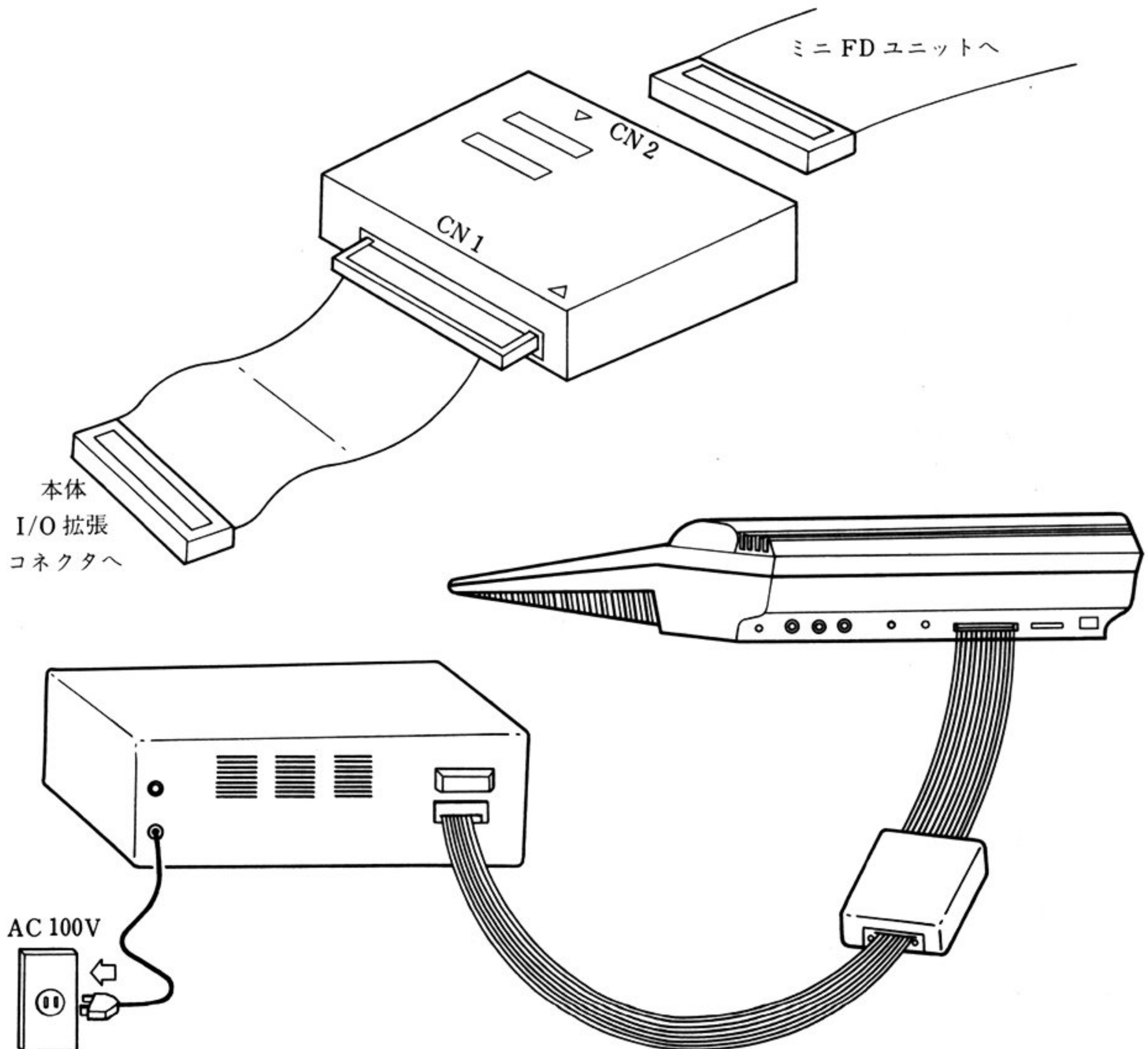
ケーブル（ミニFDユニットへ）

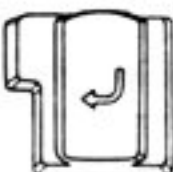
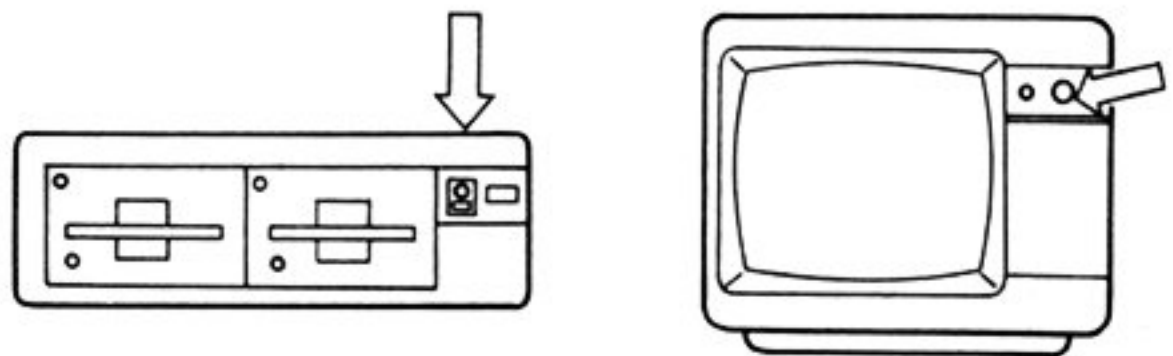
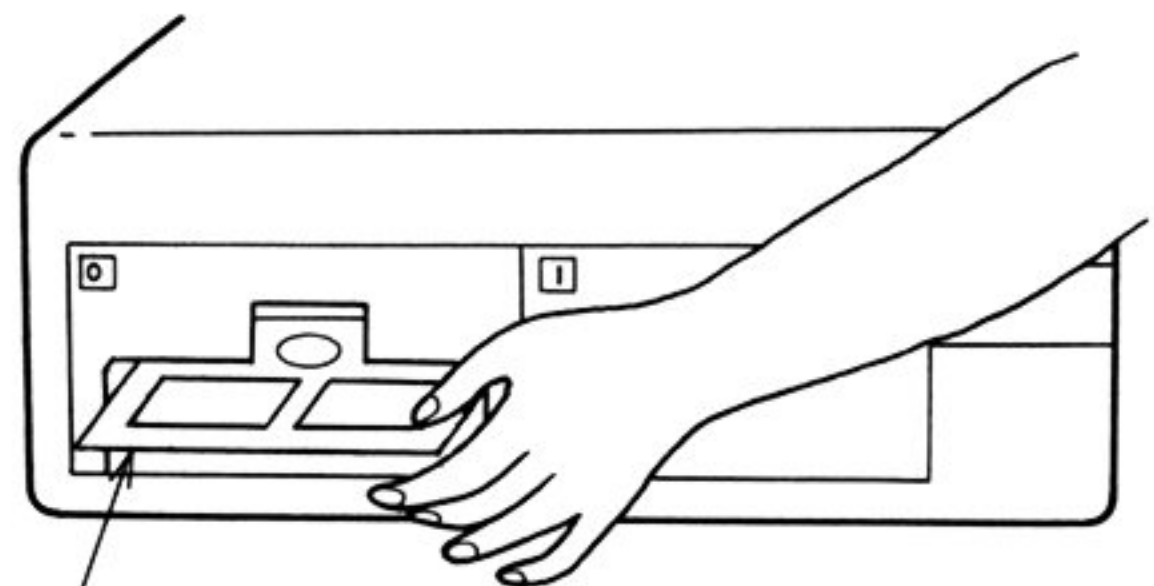
部品実装面

⑤ ケーブルを，ケーブル用溝にあわせた後，トップカバーを装着して下さい．

3.6.2 ミニフロッピィアダプタの接続

ミニフロッピィディスクユニットと，FM-8用に使われていたミニフロッピィアダプタとを接続する方法です．ミニフロッピィインタフェースカードにて，フロッピィディスクユニットを接続される方は，ミニフロッピィアダプタは必要ありませんのでご注意下さい．

① ミニフロッピィアダプタの付属品の接続ケーブルを，ミニフロッピィアダプタに接続して下さい．

② ミニフロッピィディスクユニットの背面より出ているケーブルを，ミニフロッピィアダプタに接続して下さい．

③ ミニフロッピィアダプタの接続ケーブルのもう一端を，本体背面のI/O拡張コネクタ（EXPANSION PORTの表示）に接続して下さい．

3.6.3 ミニフロッピィディスクユニットのテスト

フロッピィディスクユニットが正しく接続されているかどうかのテストを行なって下さい.

① ミニフロッピィディスクユニット, CRTディスプレイの順番で電源をONして下さい.(ミニフロッピィディスクユニットは,ヘッド保護シートを必ず抜き取ってあることを確認して下さい.)

② ミニフロッピィディスクユニットのドライブ0番(0の表示がある方)にFM-7システムディスク V3.0を挿入して下さい.(シールがはりつけてある方が上をむくようにして下さい.)ドライブカバーを閉めます.

FM-7用システムディスク

③ 本体の電源をONにして下さい.

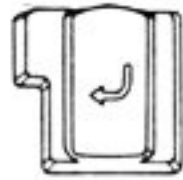
```
DISK VERSION
How many disk drives        ?■
```

④ 数秒後に, CRTディスプレイに〝ピー〟という音とともに右のような表示が表われます, ここで [↵] キーを押して下さい.

⑤ 〝ピー〟という音とともに, 次行にHow many disk files (1−15) ? という表示が表われます. ここでもやはり [↵] キーを押して下さい. 次の瞬間には右のような表示が表われます.

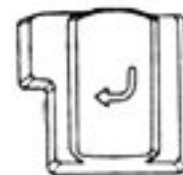
```
DISK VERSION
How many disk drives         ?
How many disk files (1-15) ?
FUJITSU F-BASIC Version 3.0
Copyright (c) 1981 By FUJITSU/MICROSOFT
×××× Bytes Free
Ready
■
```

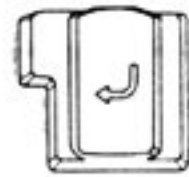
⑥ キーボードより次のように入力して下さい.

LOAD "0:DEMO1" ↵

そして，右のように表示されることを確認して下さい.

```
LOAD "0:DEMO1"
Ready
■
```

⑦ 続いて，キーボードより次のように入力して下さい.

SAVE "0:TEST" ↵

Ready 表示後に，キーボードより次のように入力して下さい.

FILES "0:" ↵

そして，"TEST" というファイル名が新たに登録されたことを，確認して下さい.

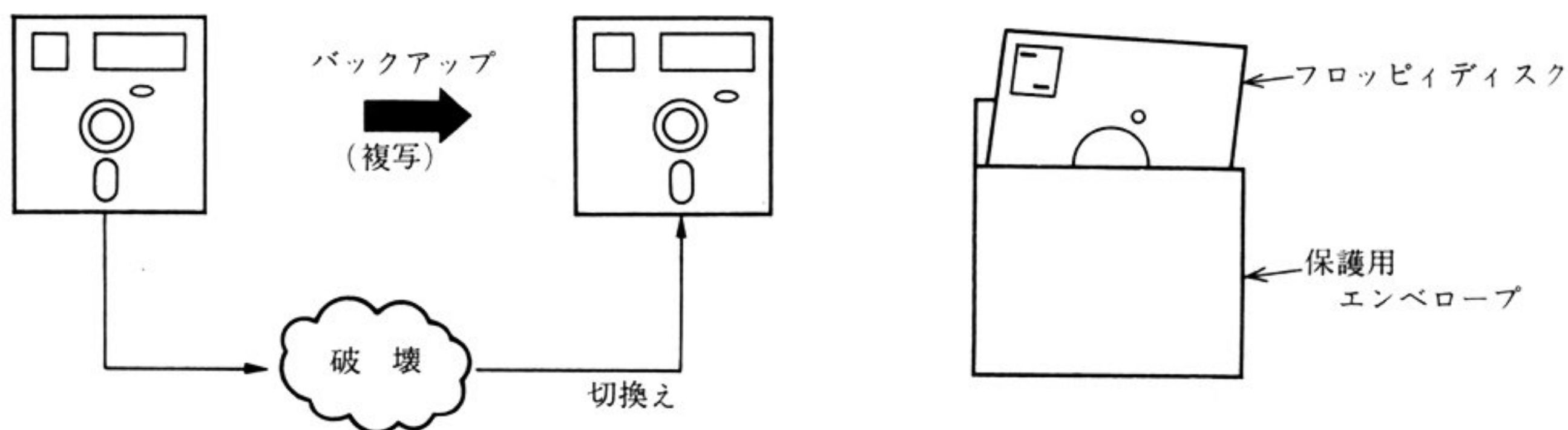
```
SAVE "0:TEST"
Ready
■
```

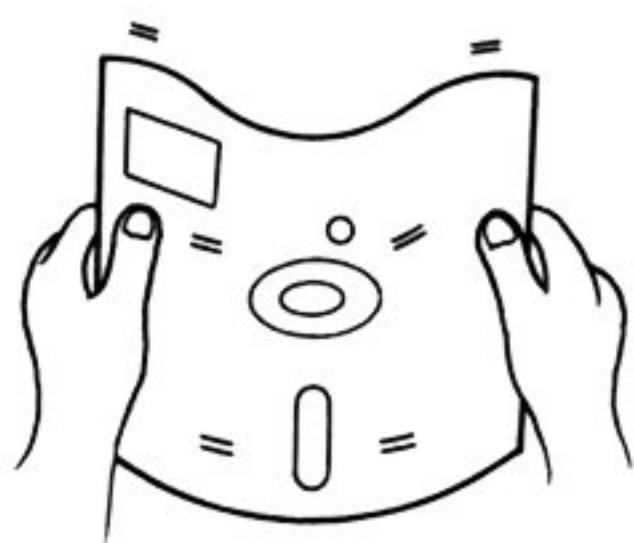
* ミニフロッピィディスクユニットに対する F-BASIC の扱い方については，第13章をお読み下さい.

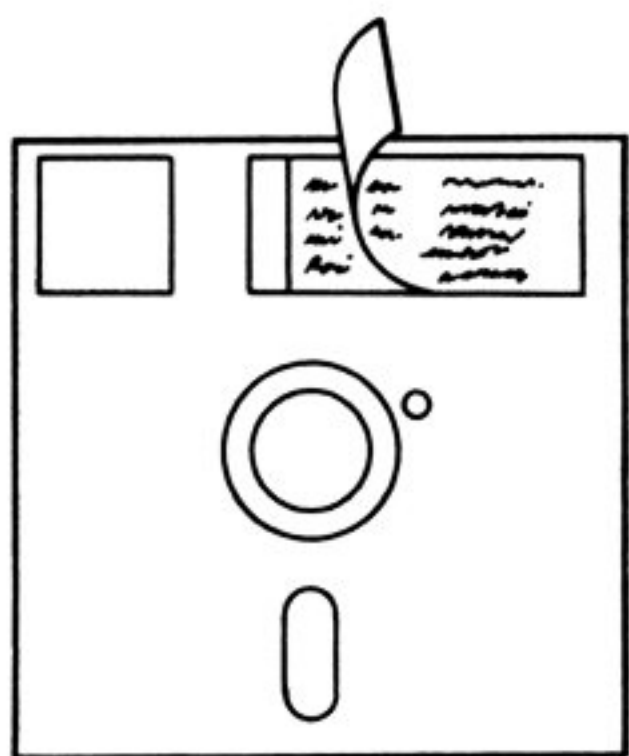
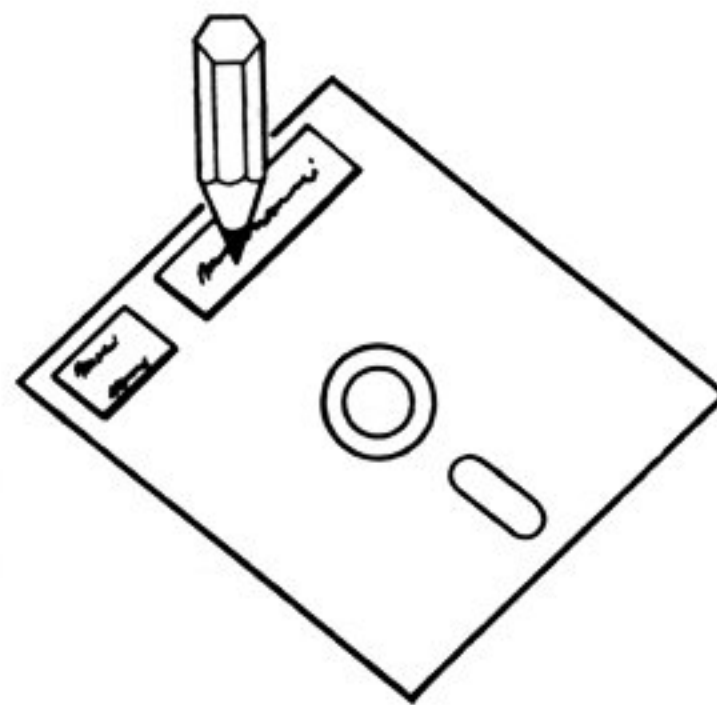
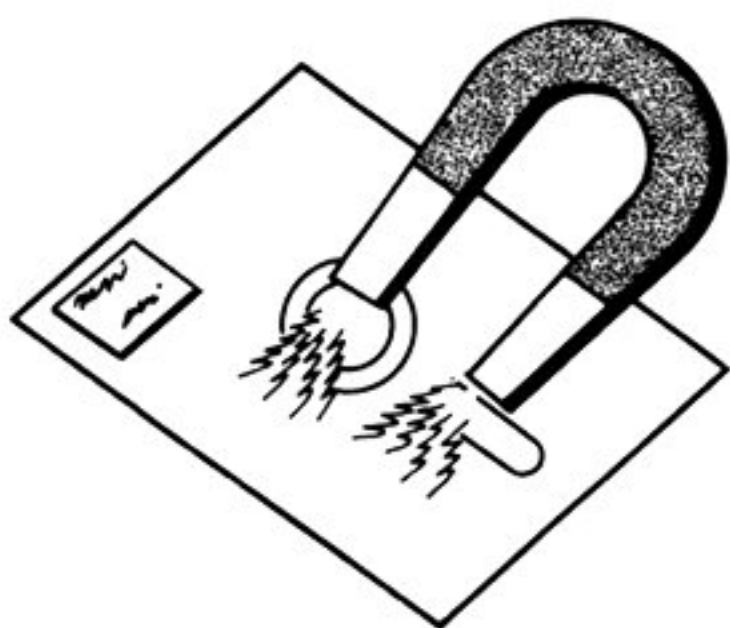
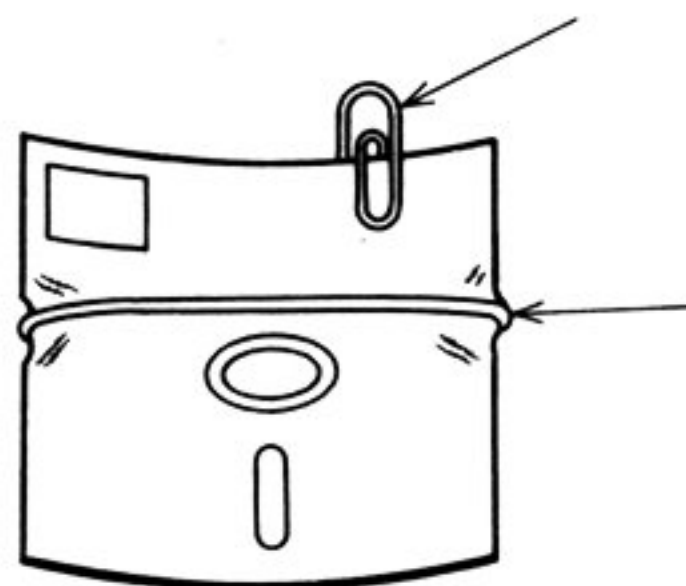
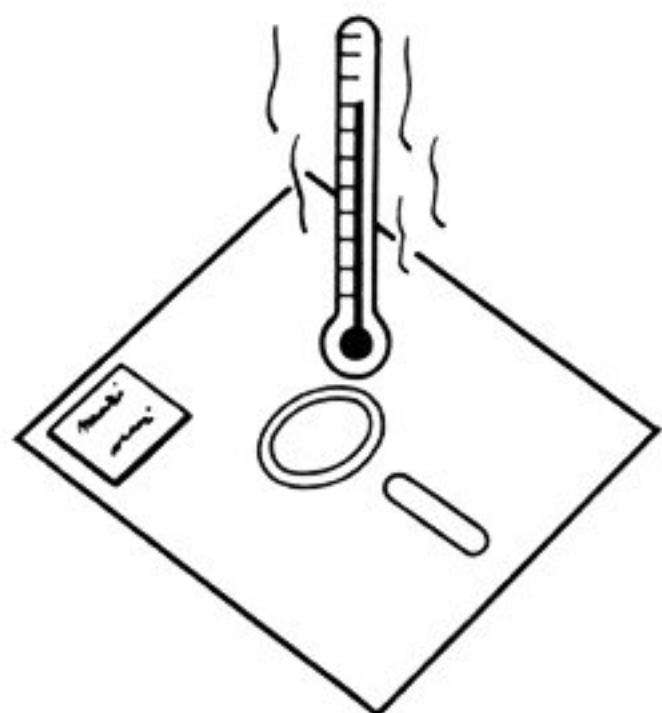
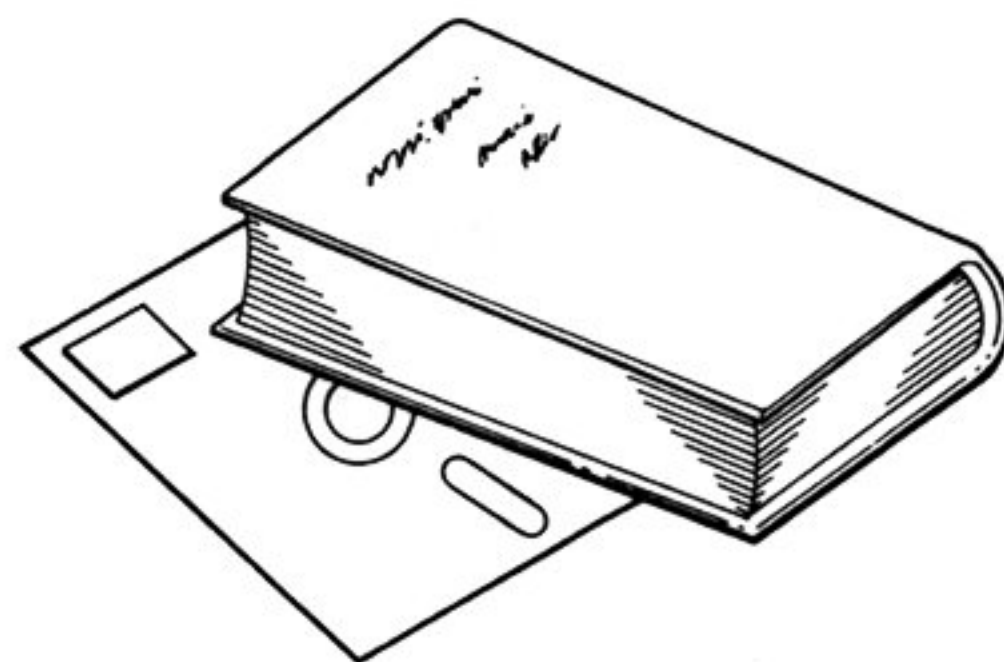
3.6.4 ミニフロッピィディスク取扱上の注意事項および仕様

（1） 取扱上の注意事項

バックアップ
（複写）
破　壊
切換え

フロッピィディスク
保護用エンベロープ

① フロッピィは，極薄，軽量で保管や取扱いやすい記録媒体ですが，限られた寿命があります．万一の場合等を考え，バックアップを必ずとって大事なデータやプログラムを守りましょう．

② フロッピィディスクの使用後は，必ず所定の保護用エンベロープに入れて下さい．保護用エンベロープは，フロッピィディスク以外のものを入れないで下さい．

③ フロッピィディスクを曲げたり折ったりしないで下さい．

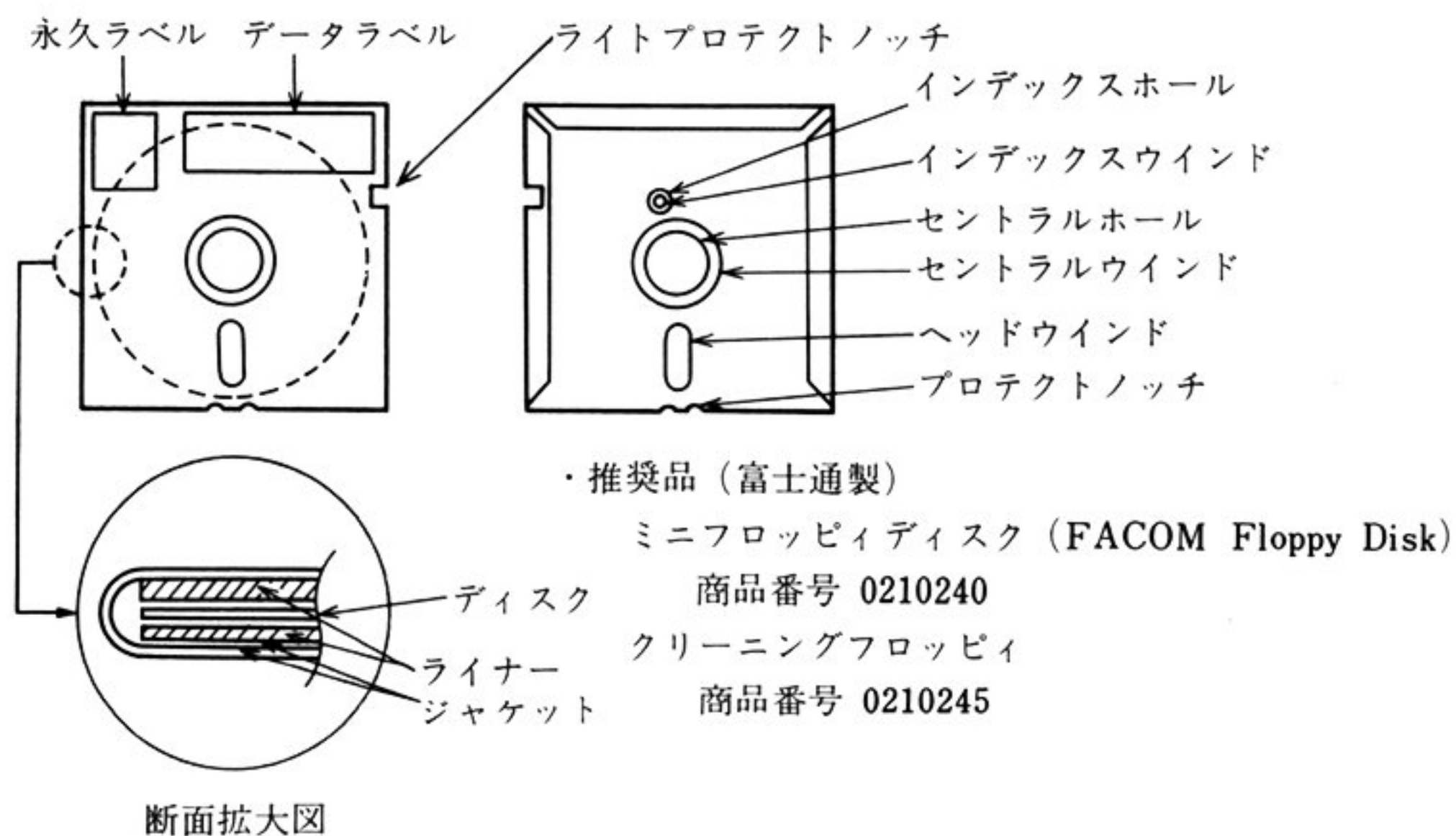
④ 食事中，喫煙中はフロッピィに触らないで下さい．

⑤ ラベルの重ね貼りはしないで下さい.

⑥ ラベルには硬筆の筆記具は用いないでフェルトペンなどを使って下さい. また, 消しゴムのかすは禁物です. 消しゴムは使わないで下さい.

⑦ 磁石を近づけないで下さい.

⑧ ゴムバンド, クリップではさまないで下さい.

⑨ 高温, 高湿な場所, 直射日光の当たる場所に置かないで下さい.

⑩ 上に物をのせないで下さい.

(2) 仕　　様

①フロッピィディスク構成図

永久ラベル　データラベル　ライトプロテクトノッチ
インデックスホール
インデックスウインド
セントラルホール
セントラルウインド
ヘッドウインド
プロテクトノッチ
ディスク
ライナー
ジャケット

断面拡大図

・推奨品（富士通製）
ミニフロッピィディスク（FACOM Floppy Disk）
商品番号 0210240
クリーニングフロッピィ
商品番号 0210245

② ミニフロッピィディスクの種類

種　　類	記憶容量
片面単密度（1S）	80Kバイト
片面倍密度（1D）	160Kバイト
両面単密度（2S）	160Kバイト
両面倍密度（2D）	320Kバイト

ミニフロッピィディスクには上の表に示すように4種類ありますが，F-BASICで使用するのは両面倍密度のものです．

③ フロッピィディスクの環境条件

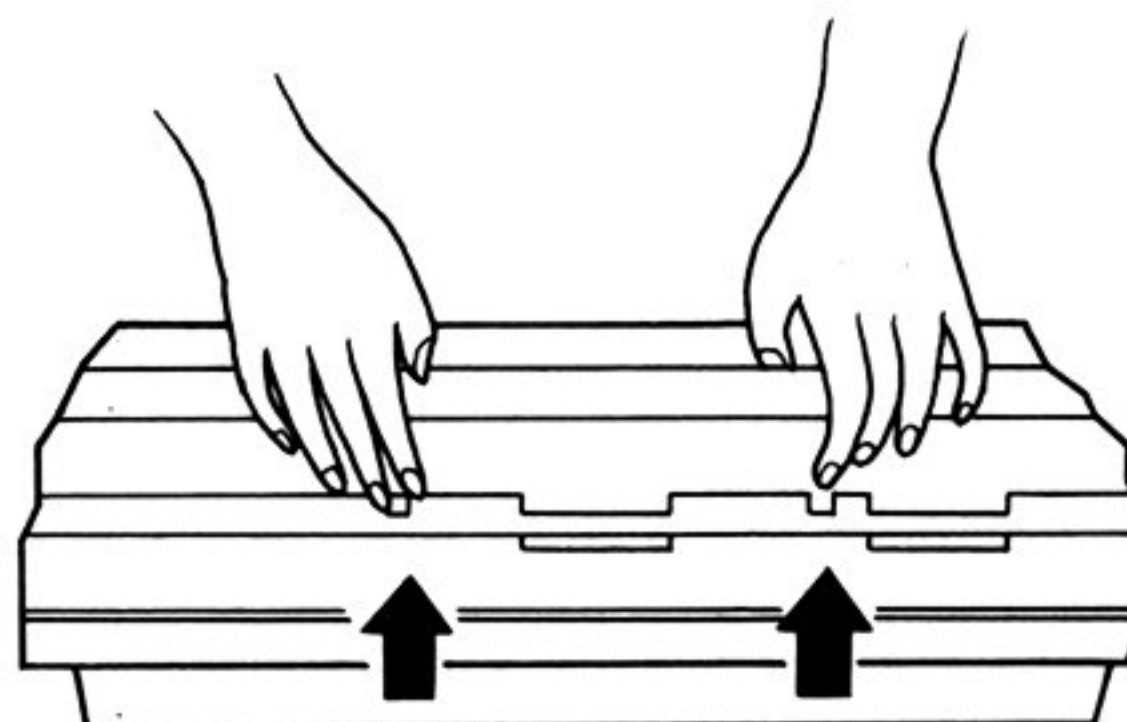
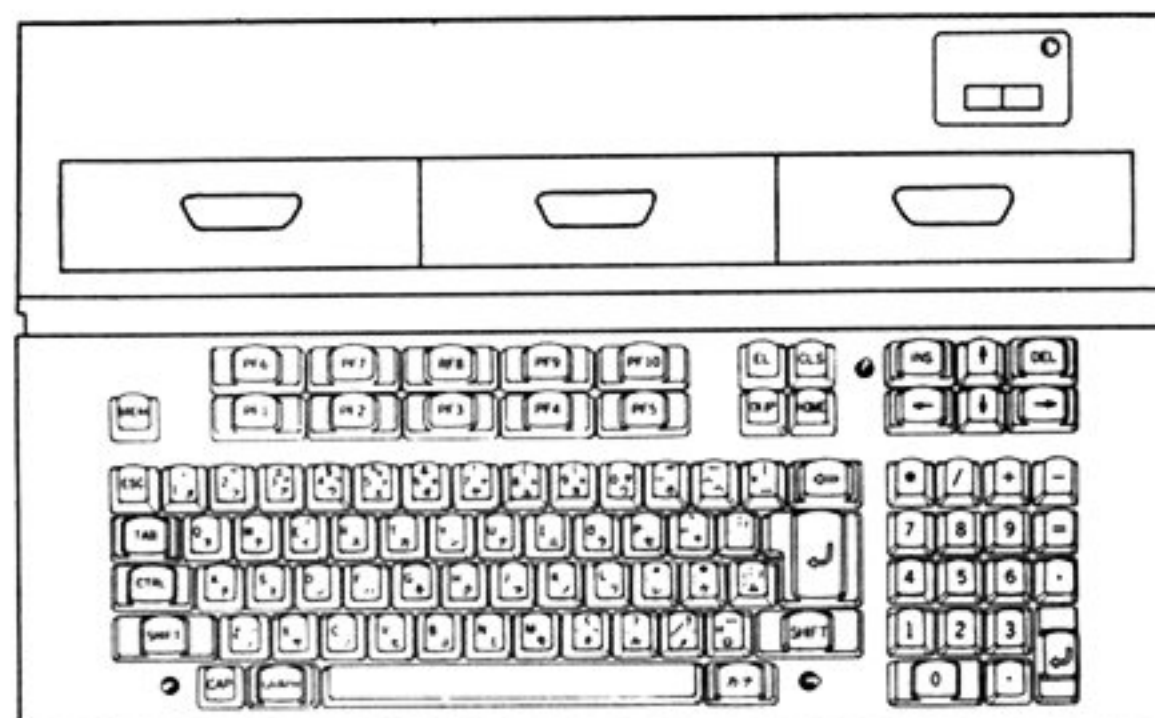
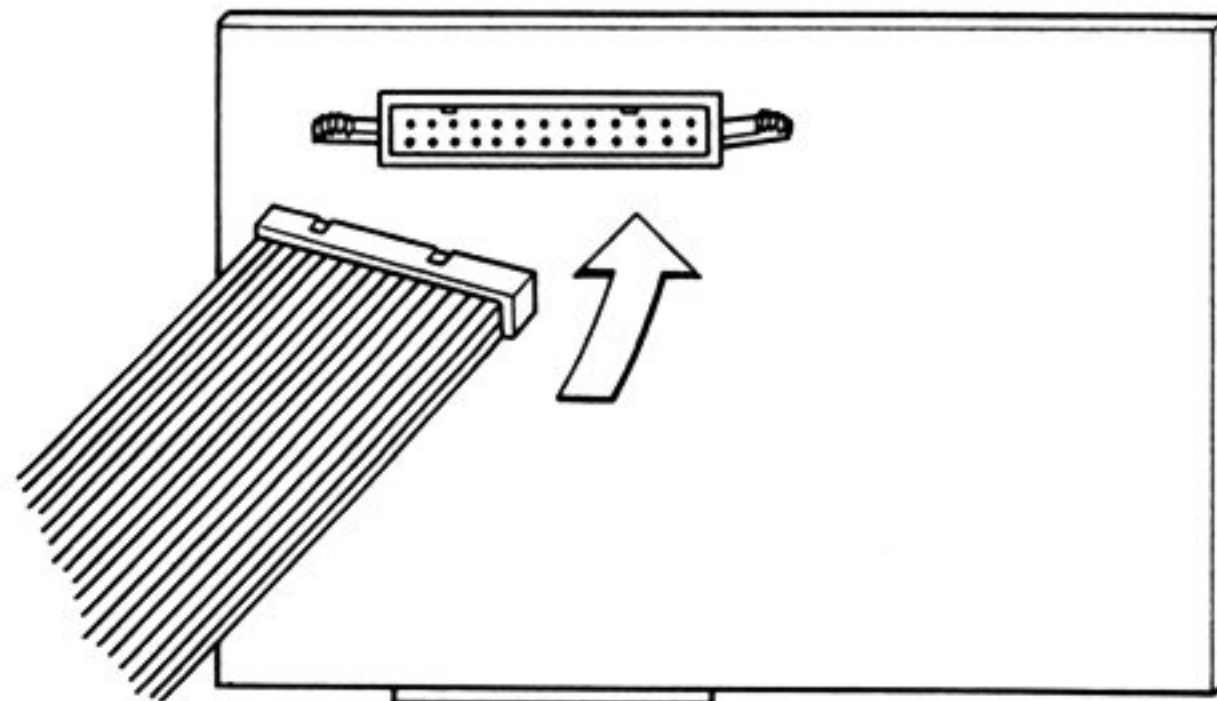
	動作時	保管時	運搬時
温　度	4～52℃	4～53℃	−40～53℃
湿　度	8～80％RH（最大湿球29.4℃）		8～90％RH（結露なし）
温度勾配	15℃/H以下		
外部磁界	4,000 A/m (50エルステッド）以下		

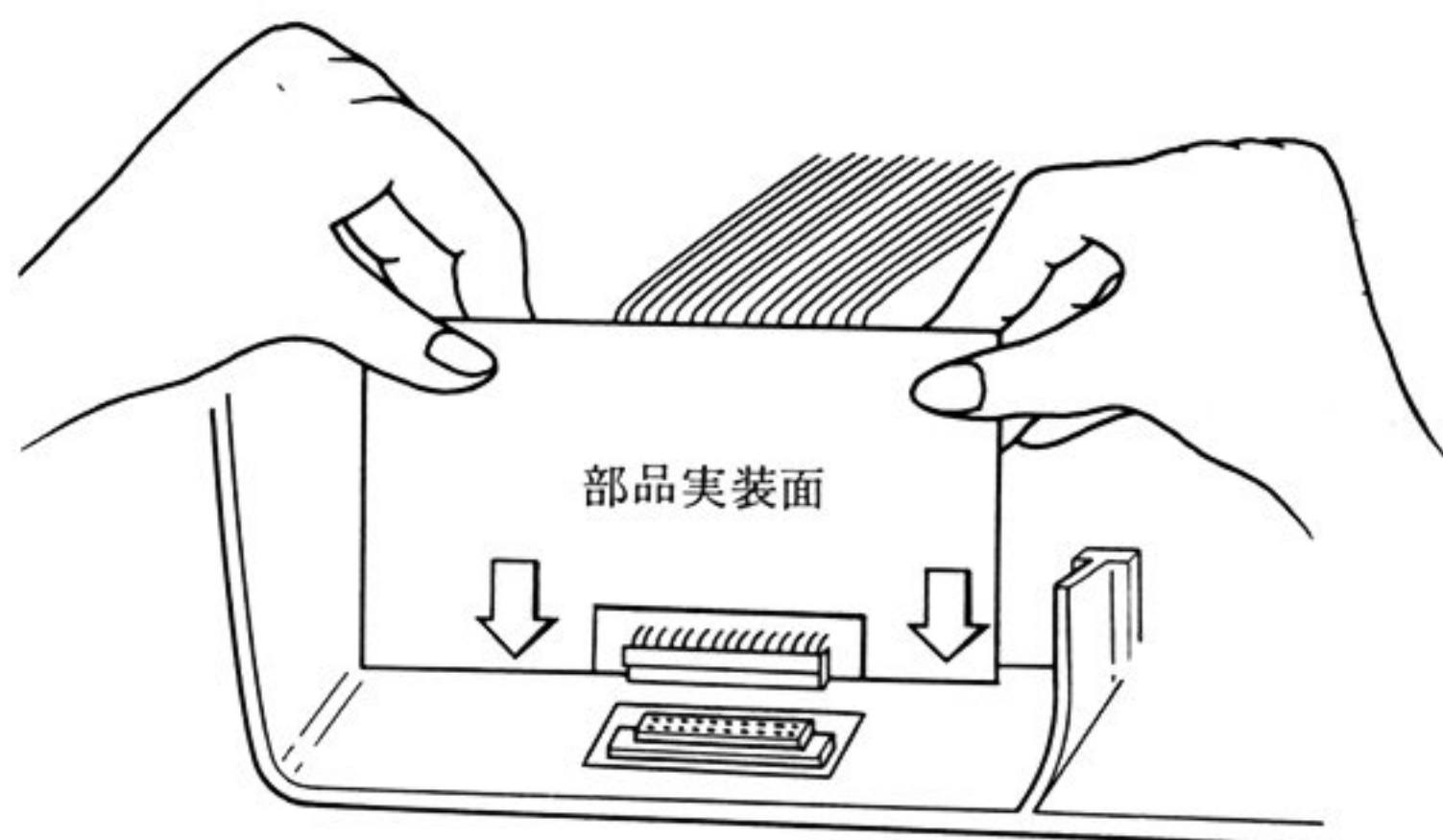
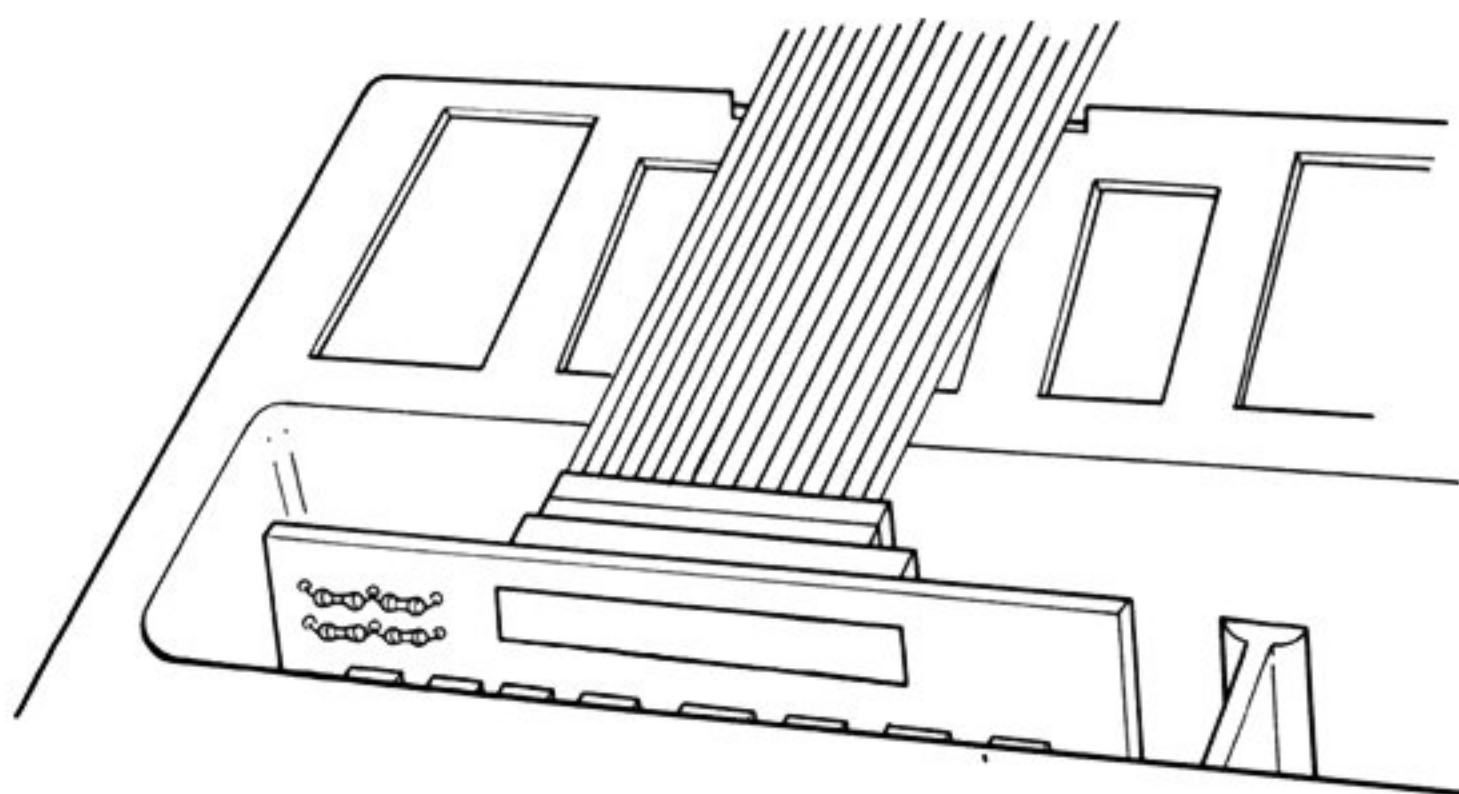
3.7 オプションカード

オプションカードは以下の3種類があります.

漢字ROMカード	このカードを装着することにより，JIS第一水準(2965文字)の漢字，ひらがな，カタカナ，ローマ字，ギリシャ文字，ロシア文字が表示できます.
ミニフロッピィインタフェースカード	ミニフロッピィディスクユニットと本体を接続する際に必要となります.
RS-232C インタフェースカード	RS-232Cインタフェースを持つ周辺装置や，音響カプラ，モデムなどを使用し，FM-7同士などを接続して使う場合に必要となります.

〈装着方法〉

① 本体上部のトップカバーをはずします.
カバー後部にある固定爪を押しながら，ゆっくりカバーをはずします.
固定爪の取扱いはていねいに行なって下さい.

② カバーを開けますと，コネクタが3つ並んでいます．左側，中央のいずれかにセットします.

③ RS-232Cインタフェースカード，ミニフロッピィインタフェースカードは，先に接続用のケーブルを接続した後，装着します．

④ 本体にカードを装着します．部品実装面を手前側にしてコネクタに注意しながら接続して下さい．

部品実装面

⑤ RS-232Cインタフェースカード，ミニフロッピィインタフェースカードの場合は，ケーブルをケーブル用溝にあわせた後，トップカバーを装着して下さい．

第4章　システムの起動

4.1　システムの起動

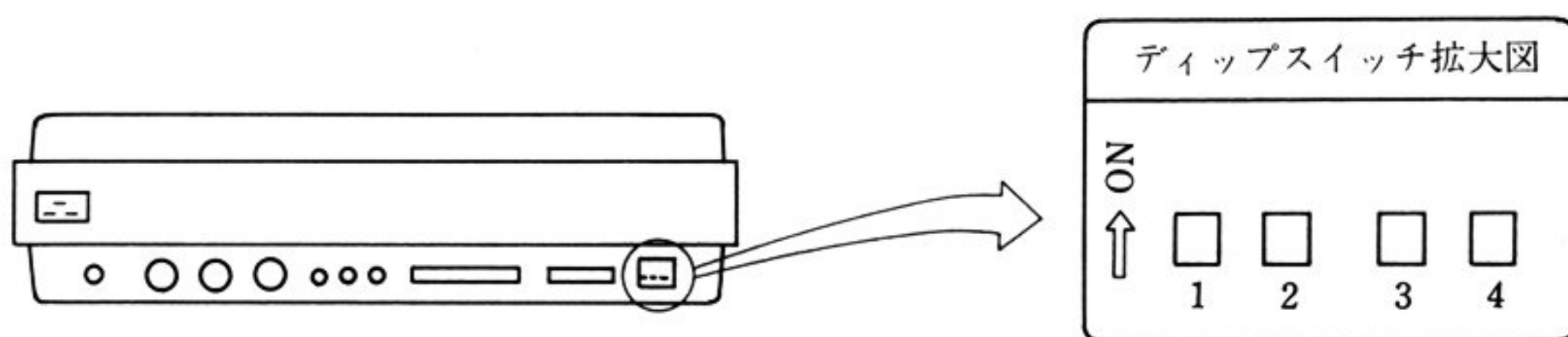
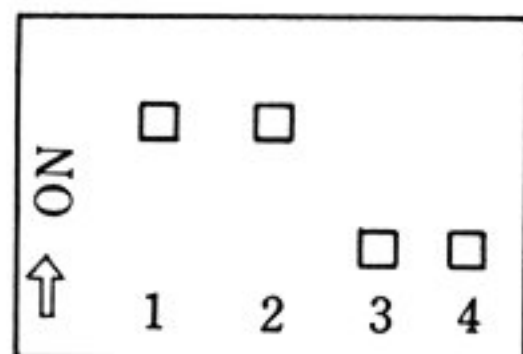
4.1.1　ディップスイッチ

本体背面にディップスイッチがあります．1番と2番がブートローダ用に,3番は未使用,4番はクロック周波数設定用になっています．

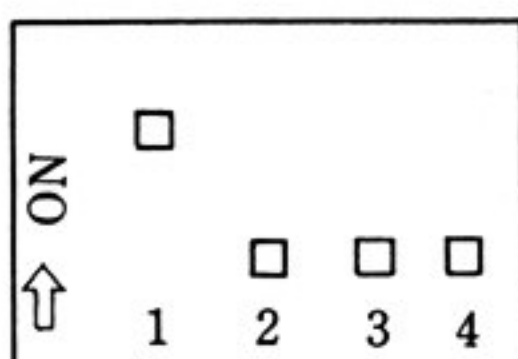
ディップスイッチ拡大図

ON
1　2　3　4

(1)　ROMモード/DISKモード

ON
1　2　3　4

F-BASIC (ROMモード/DISKモード) が起動されます．ROMモードになるかDISKモードになるかの区別は，電源投入時あるいはリセット時にシステムディスクがドライブユニット0番にセットされて，レディ状態にあるか否かによって決まります．したがって，いったんROMモードにある状態でも，システムディスクをセットしてリセットすればDISKモードになります．また逆も成立ちます．

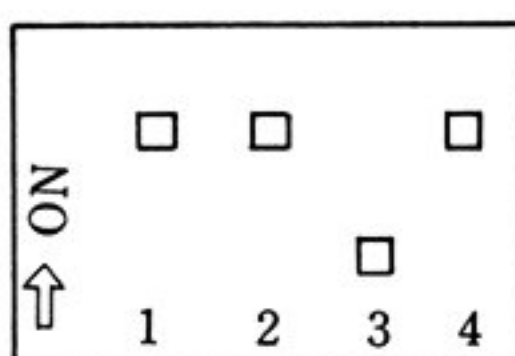
(2) DOSモード

ON ⇧ 1 2 3 4

1番をON, 2番をOFF, 3番をOFFに設定し，電源をONにしますと，ドライブユニット0番にセットされたミニフロッピィディスク内のシステムを起動することができます.

たとえば，デップスイッチをこのように設定し，ドライブユニット0番にOSの格納されたシステムディスクをセットし，電源をONにしますとフロッピィディスクに格納されていたOSが起動します.

3番は使用しておりませんので，常にOFF状態とします.

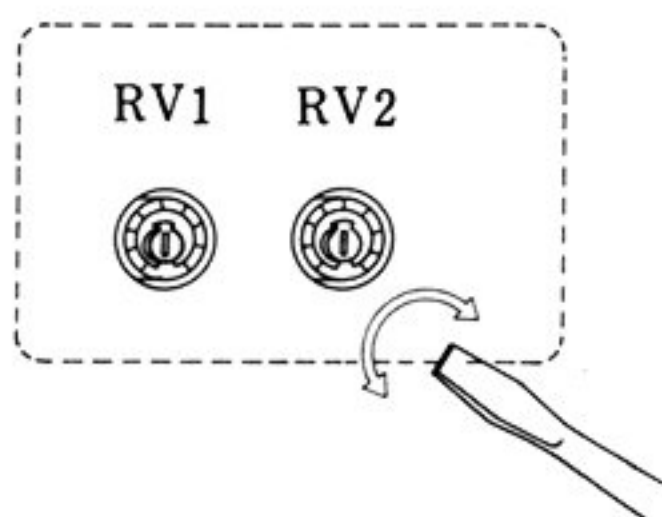
(3) クロック周波数設定

ON ⇧ 1 2 3 4

4番のスイッチのON･OFFの切換えにより，クロック周波数の設定を行ないます.

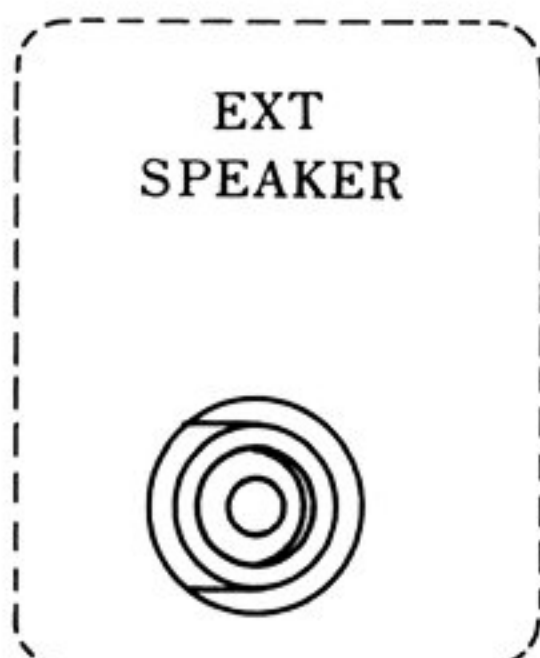
スイッチの設定	メイン CPU		サブ CPU	
	クロック周波数	動作周波数	クロック周波数	動作周波数
ON	4.9 MHz	1.2 MHz	4.0 MHz	1.0 MHz
OFF	8.0 MHz	2 MHz	8.0 MHz	2.0 MHz

4.1.2 ブザーボリューム，サウンドボリュームの調整

ブザー音，サウンド音の音量の調整ができます．出荷時には，あらかじめ音量の調整はされていますが，本機を設置する場所や好みに合わせて，調整を行なって下さい.

RV1 RV2

本体背面にRV1(サウンドボリューム調整)，RV2(ブザーボリューム調整）の表示のボリュームがありますので，ここをドライバーにて左右に回して調整を行なって下さい．左側に回しますと，音は小さく，右側に回しますと音は大きくなります.

4.1.3 外部スピーカ用コネクタ

EXT
SPEAKER

この外部スピーカ用コネクタに，市販のスピーカを接続することにより，サウンド機能で作り出した音楽をより楽しむことができます.

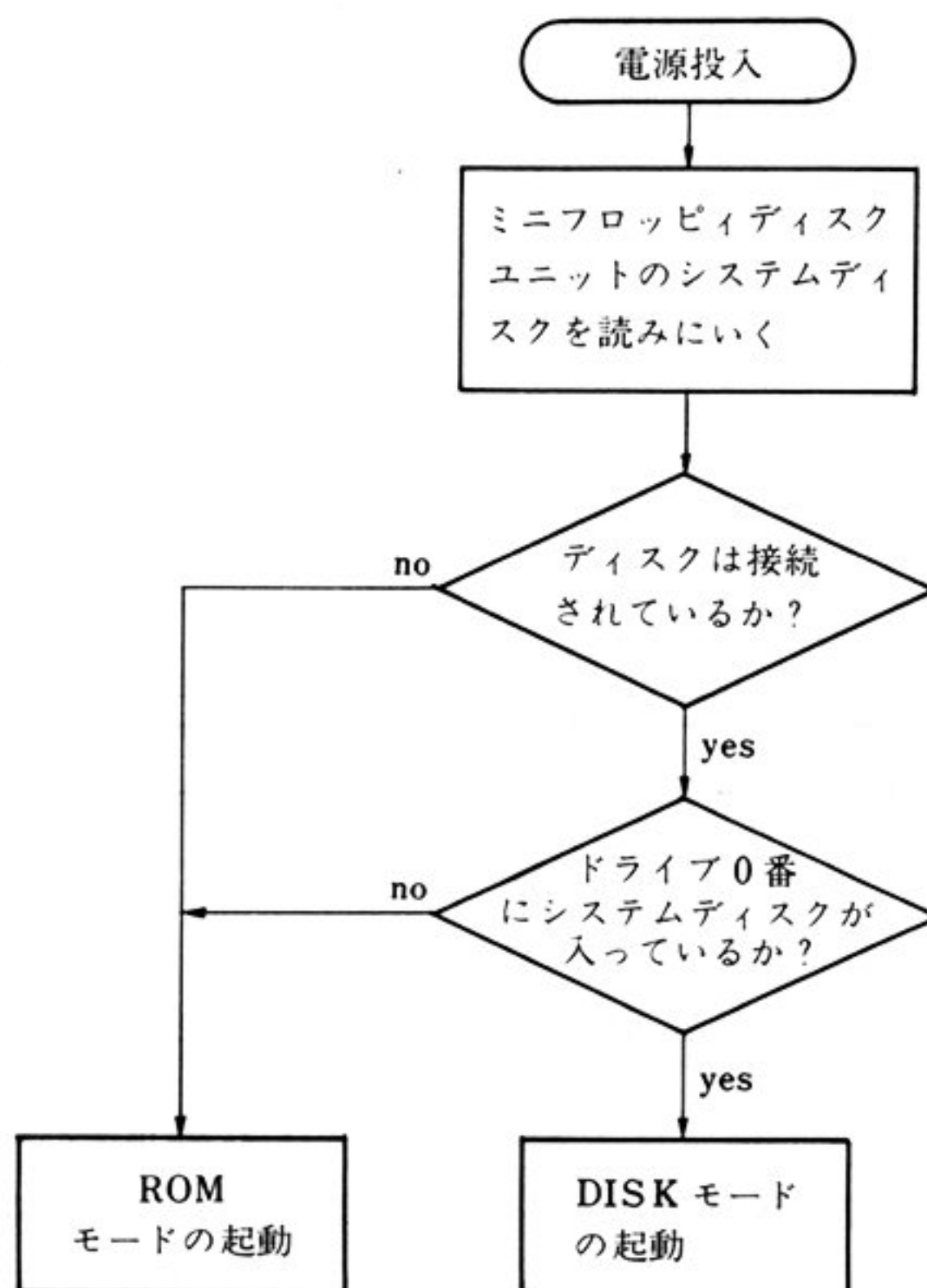
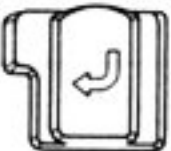
音量の調整は本体背面のサウンドボリューム（RV1の表示）により行なって下さい．接続する市販のスピーカは入力インピーダンスが4Ω～16Ωのものをご使用下さい.

4.2 F-BASIC の起動

F-BASIC には，主に本体内蔵の BASIC ROM を使用する ROM モードと，F-BASIC システムディスク内の DISK BASIC と，本体内蔵の BASIC ROM の一部を使用した DISK モードの 2 種類に分けられます．DISK モードの F-BASIC は，フロッピィディスクを取扱う際に使用します．ROM モードの起動をかけた場合には，フロッピィディスクよりプログラムなどを読込んだり，またフロッピィディスクにプログラムなどを格納することはできませんので，フロッピィディスクを取扱う場合には，必ず DISK モードの起動をかけなければなりません．

F-BASIC の種類

バージョン	媒体	名称	略称	適用機種
V1.0	FM-8 本体の ROM	F-BASIC V1.0 ROM モード	V1.0/ROM	FM-8
	FM-8 本体内の ROM およびミニフロッピィディスクに添付されているシステムディスク	F-BASIC V1.0 DISK モード	V1.0/DISK	
V2.0	標準フロッピィディスク	F-BASIC V2.0 8インチ	V2.0/8	FM-8
V3.0	FM-7 本体内の ROM	F-BASIC V3.0 ROM モード	V3.0/ROM	FM-7
	FM-7 本体内の ROM および V3.0 用のシステムディスク	F-BASIC V3.0 DISK モード	V3.0/DISK	

電源投入
↓
ミニフロッピィディスクユニットのシステムディスクを読みにいく
↓
ディスクは接続されているか？ — no → ROM モードの起動
↓ yes
ドライブ0番にシステムディスクが入っているか？ — no → ROM モードの起動
↓ yes
DISK モードの起動

4.2.1 F-BASIC（DISK モード）の起動

DISK モードは，フロッピィディスクを取扱う時に使用する BASIC です．DISK モードの起動は次のような手順で行なわれます．

(1) ディスプレイ，ミニフロッピィディスクユニットなど周辺装置の電源を入れて下さい．

(2) ミニフロッピィディスク，ドライブユニット 0 番に FM-7 用システムディスクをセットして下さい．

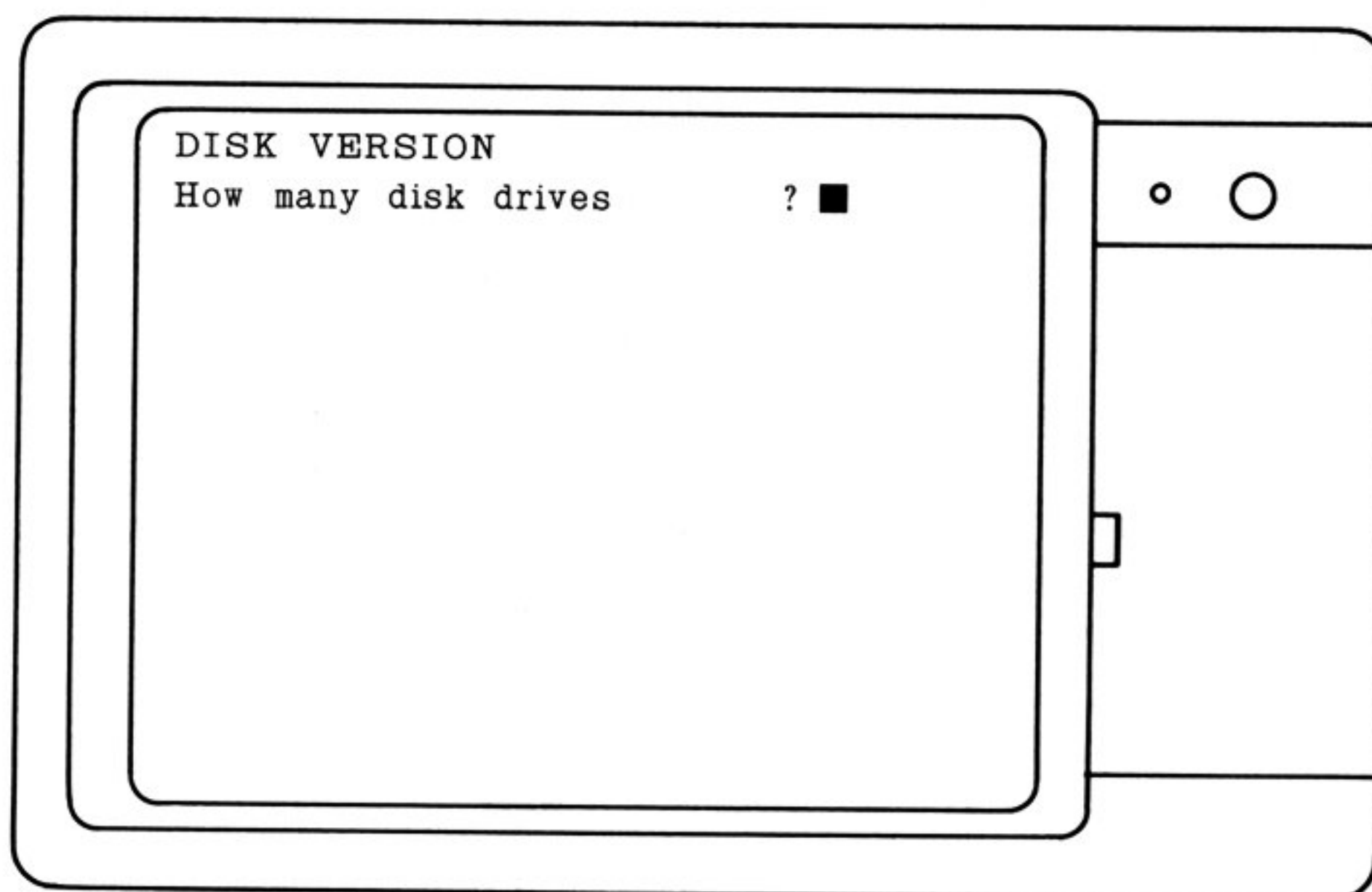
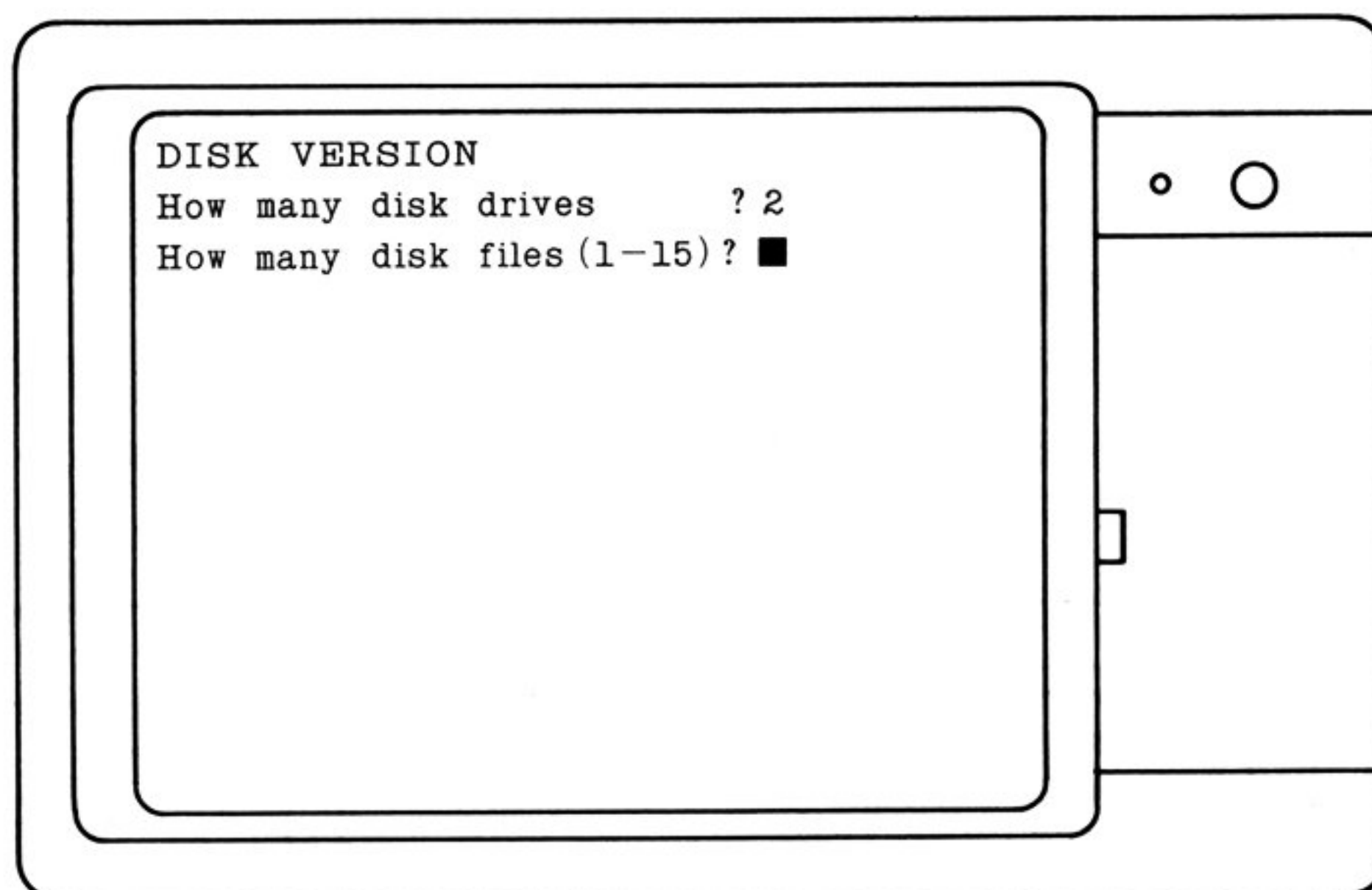
(3) 本体の電源を入れて下さい．すでに電源が入っている場合には，リセットスイッチを押して下さい．

(4) 数秒カーソルが点滅した後に“ピー”という音とともに次のような表示があり使用するドライブ数を質問してきますので，使用するドライブ数を入力します（**画面 1**）．

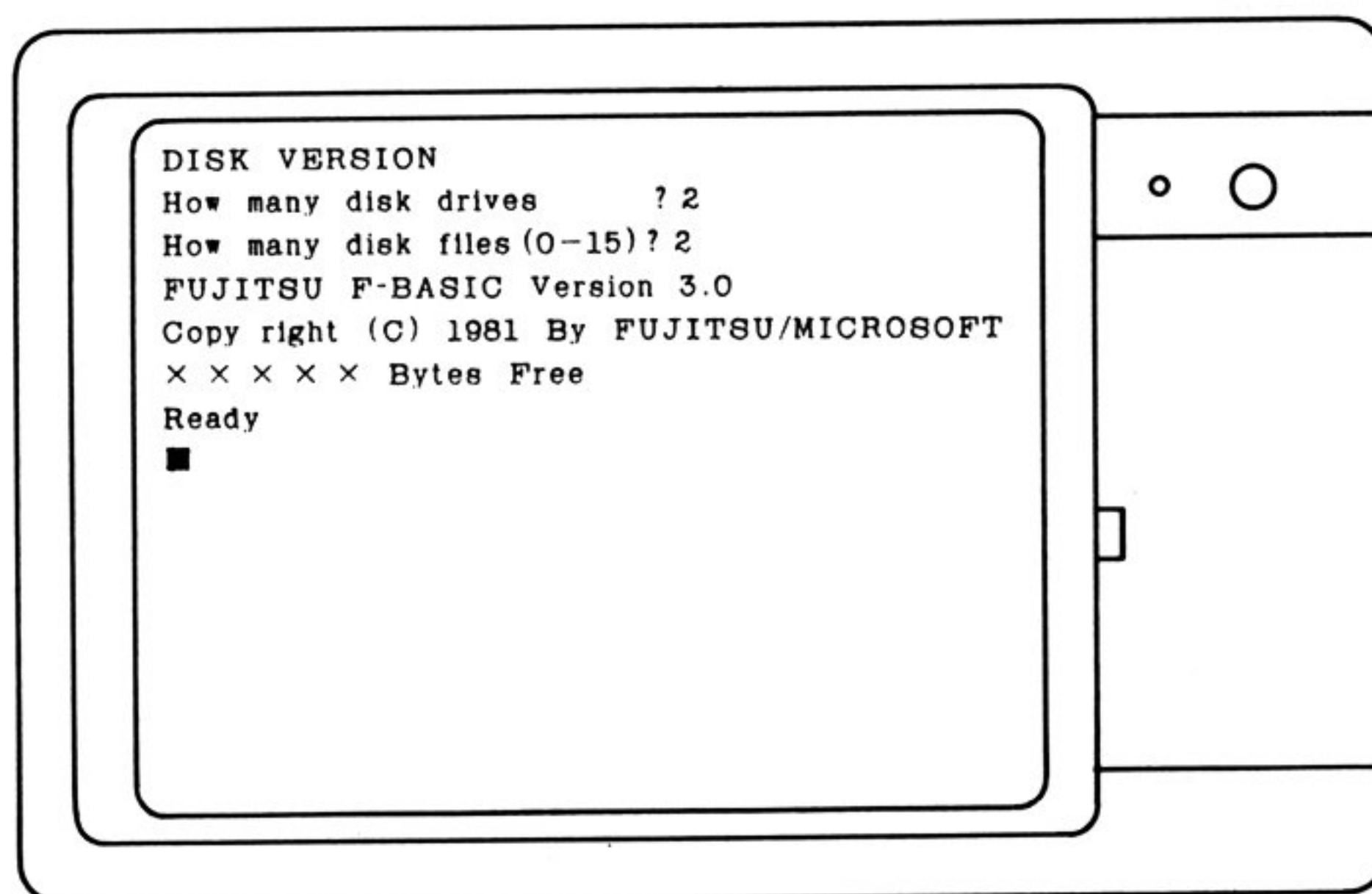
1 ドライブ使用：［1］，［↵］の順番でキーを入力します．

2 ドライブ使用：［2］，［↵］の順番でキーを入力します．

省　略　形：［↵］のみ入力しますと，2 ドライブ使用と設定されます．

```
DISK VERSION
How many disk drives        ? ■
```

画面 1

(5) 使用ドライブ数の入力を終えますと，また“ピー”という音とともに次の表示になり（**画面 2**），1 つのプログラム中で同時に OPEN されるフロッピィディスク上のファイルの数量を質問してきます（ファイルについての説明は第 11 章ファイルをお読み下さい）．同時に OPEN できるフロッピィディスク上のファイルの数は 15 ファイルまでです．0〜15 の範囲で使用するファイル数を入力し

て下さい．ファイル数の入力を省略し[↵]キーを押しますと，同時にOPENされるファイルの数は1ファイルと設定されます．

```
DISK VERSION
How many disk drives      ? 2
How many disk files(1−15)? ■
```

画面2

(6) 使用ファイル数の入力を終えますと，**画面3**のような表示があり，Ready状態（コマンド受けつけ状態）になり，DISKモードのBASICが使用できます．この画面では，2ドライブ，ディスクを使い，同時にOPENするファイルの数を2ファイルと設定した画面です．×××× Bytes Freeと表示してある所は，使用ドライブ数，使用ファイル数により異なります．

```
DISK VERSION
How many disk drives      ? 2
How many disk files(0−15)? 2
FUJITSU F-BASIC Version 3.0
Copy right (C) 1981 By FUJITSU/MICROSOFT
× × × × × Bytes Free
Ready
■
```

画面3

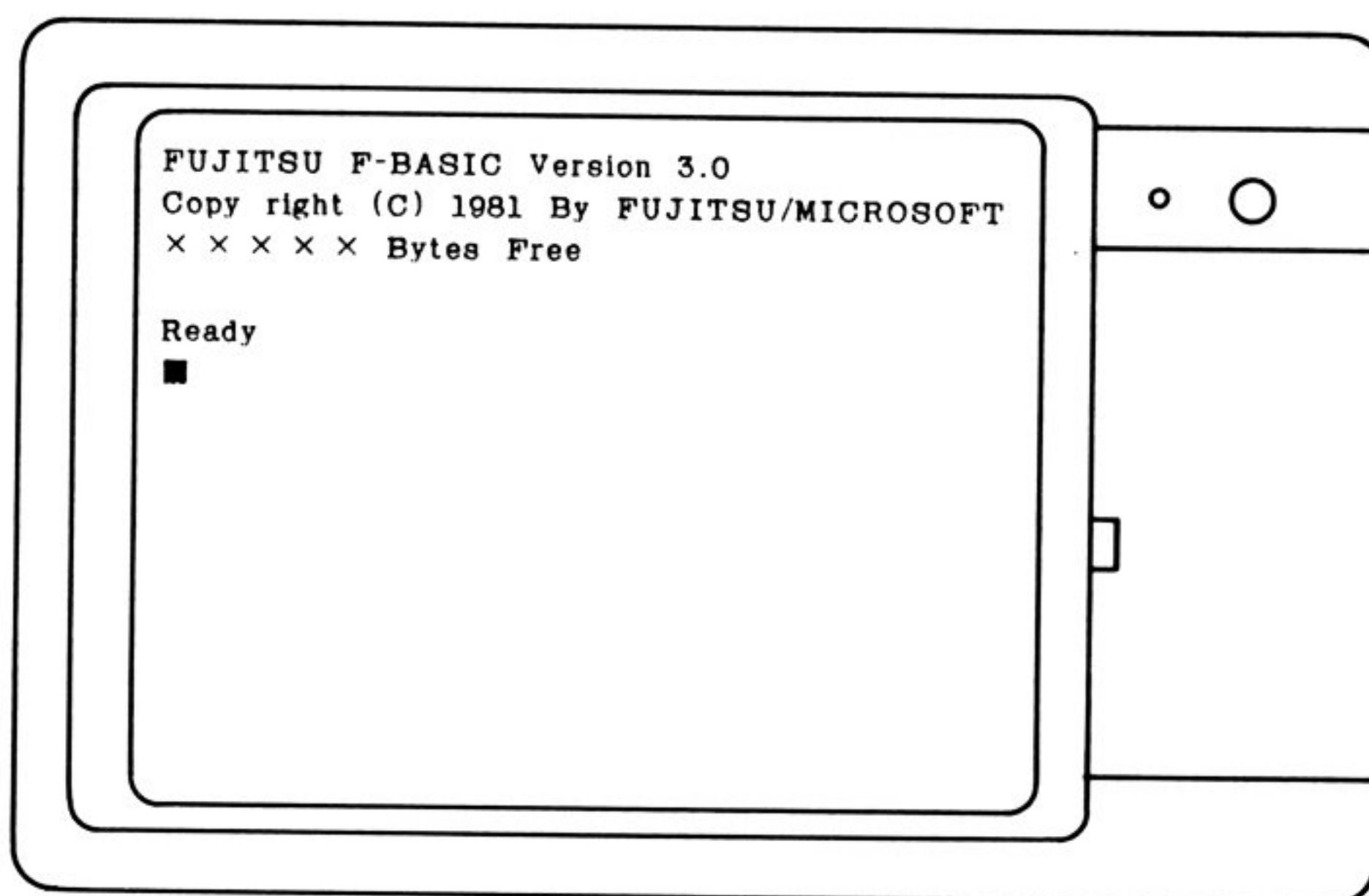
4.2.2 F-BASIC（ROMモード）の起動

ROMモードとは，本体に実装されているBASIC ROMの中に入っているBASICに許される命令を使うものです．主に，カセットテープを補助記憶装置として扱う場合に使用します．

ROMモードの起動がかかる状態は，ミニフロッピィディスクユニットを接続していない時，またはフロッピィディスクユニットのドライブ0番にFM-7用システムディスクが入っていない時です．

(1) ディスプレイなど周辺装置の電源をONにして下さい．

(2) 本体の電源を入れて下さい．数秒しますとディスプレイに次のような表示（**画面4**）があり，Ready状態（コマンド受付状態）となり，ROMモードのBASICが使用できます．

```
FUJITSU F-BASIC Version 3.0
Copy right (C) 1981 By FUJITSU/MICROSOFT
× × × × × Bytes Free

Ready
■
```

画面4

4.3 FM-7で扱えるDOS

FM-7にて使用できる流通のDOS(ドス:Disk Operating System)は以下の通りです．これらを使用する際は，ディップスイッチは次のように設定して下さい．

ON↑ 1 2 3 4　または　ON↑ 1 2 3 4

名　称	内　　　　容	必要なハードウェア
UCSD Pascal	構造化プログラミング，豊富なデータ構造を特長として持つPascal言語を，マイクロコンピュータ用にUCSD(カリフォルニア大学サンディエゴ校)が開発したOS付言語です．操作は会話形式で行ないます．また，UCSD Pascalは強力なスクリーンエディタを備えています．	本　体 ミニフロッピィディスクユニット ディスプレイ
FLEX	FLEXは68系CPUのために作られた汎用DOSです．シングルユーザークラスのOSとして多機能で操作性もすぐれています．TSC社をはじめ，数社から発売されている豊富なソフトウェアを利用できます．	本　体 ミニフロッピィディスクユニット ディスプレイ

4.3 FM-7で扱える DOS

FM-7にて使用できる流通のDOS（ドス：Disk Operating System）は以下の通りです．これらを使用する際は，ディップスイッチを右図のように設定して下さい．

ON ↑ 1 2 3 4　または　ON ↑ 1 2 3 4

名　称	内　　　　容	必要なハードウェア
UCSD Pascal	構造化プログラミング，豊富なデータ構造を特長として持つ Pascal 言語を，マイクロコンピュータ用に UCSD (カリフォルニア大学サンディエゴ校)が開発したOS付言語です．操作は会話形式で行ないます．また，UCSD Pascal は強力なスクリーンエディタを備えています．	本　体 ミニフロッピィディスクユニット ディスプレイ
FLEX	FLEXは68系CPUのために作られた汎用DOSです．シングルユーザークラスのOSとして多機能で操作性もすぐれています．TSC社をはじめ，数社から発売されている豊富なソフトウェアを利用できます。	本　体 ミニフロッピィディスクユニット ディスプレイ
CP/M	Digital Reserch 社により初めてパーソナルコンピュータ用に実用化されたシングルユーザタスクのOSです．パーソナルコンピュータ用OSとしては最高クラスの拡張機能を誇り，ほとんどすべてのアプリケーションに対して強力な機能を発揮します．	本　体 ミニフロッピィディスクユニット ディスプレイ Z80カード

4.4 フロッピィディスクの取扱い

4.4.1 取扱い方

フロッピィディスクは，信頼性の高い補助記憶媒体ですが，以下のことにより，記録されている内容や，媒体を破壊することがあります．

・操作およびプログラムミス

・フロッピィディスクの物理的な損傷

・ハードウェアのトラブル

システムディスクを破壊しますと，F-BASIC V3.0 DISKモードの起動ができなくなります．

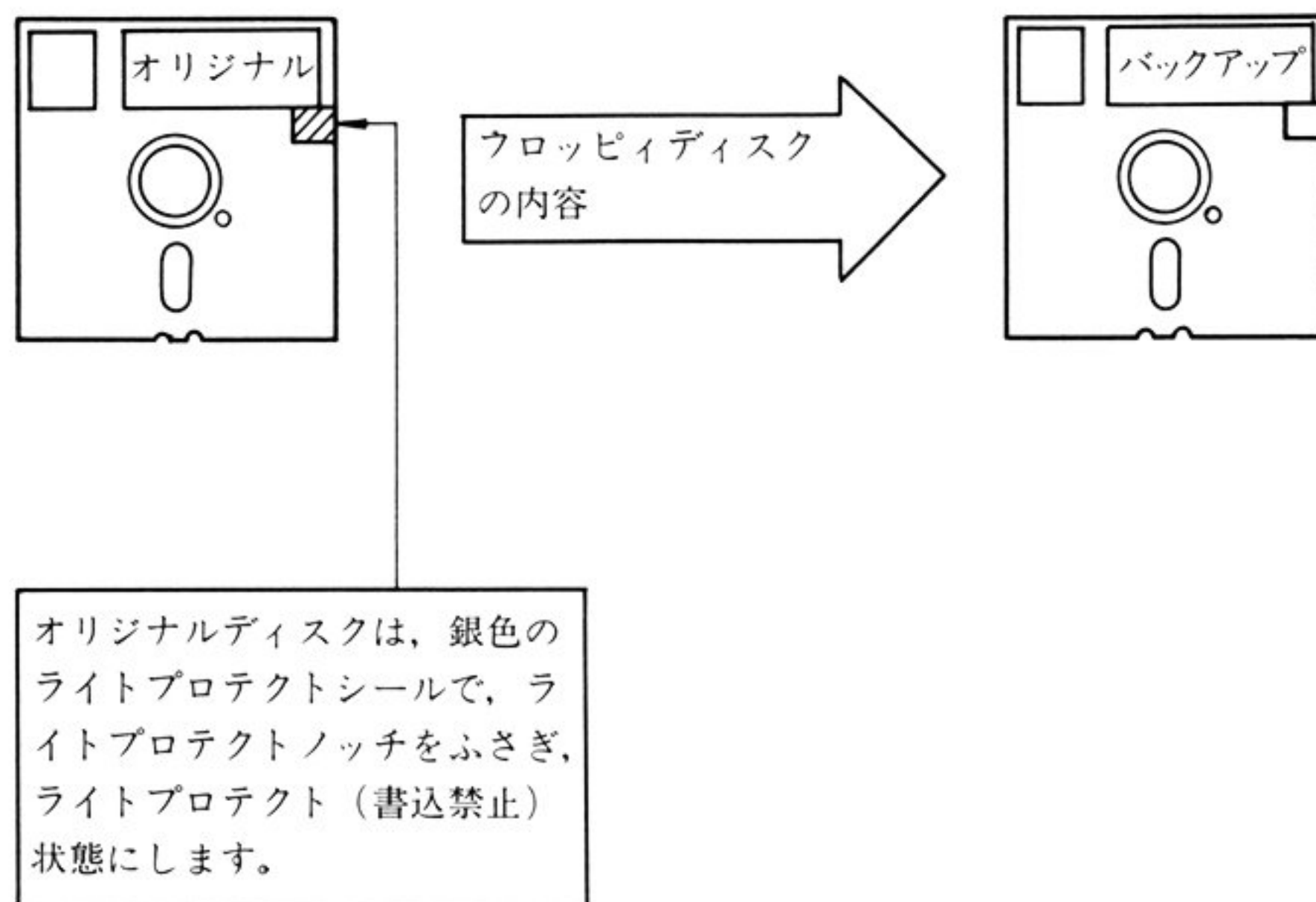
上記のような思わぬ事故から，フロッピィディスクを守るため，常に，フロッピィディスクの内容を全てコピーした，バックアップディスクを作成して下さい．

バックアップディスクに対して，コピーを行なうフロッピィディスクを，オリジナルディスクとよびます．

オリジナル

フロッピィディスク
の内容

バックアップ

オリジナルディスクは，銀色の
ライトプロテクトシールで，ラ
イトプロテクトノッチをふさぎ，
ライトプロテクト（書込禁止）
状態にします。

バックアップディスクを常に使用し，オリジナルディスクは，バックアップディスクが使用不能となったときのために，大切に保管します．

市販のフロッピィディスクを，F-BASICにて扱うためには，FM-7システムディスクにより，以下の処理が必要です．

・フォーマッティング（記録形式）を与える処理

・DISKコードのコピーを行ない，システムの起動を可能とする処理

（DISKコードとは，F-BASICのDISKを扱う命令のことです.）

記録形式や，DISKコードを持つフロッピィディスクを，システムディスクとよびます。

また，システムディスクへ，プログラム等を記録し，使用者（ユーザ）が通常使用するフロッピィディスクを，ユーザディスクとよびます．

次に，各フロッピィディスクの扱い方を示します．

〈フロッピィディスクの扱い方〉

フロッピィディスクの種類	FM-7 システムディスク	システムディスク	ユーザディスク
扱い方	バックアップディスクの作成	システムディスクの作成	バックアップディスクの作成
使用目的	システムの起動，ディスクユーティリティの使用	システムの起動，プログラムやデータの作成および記録	アプリケーションプログラムの使用
作成するために使用するディスクユーティリティ	SYSDSK VOLCOPY	SYSDSK	SYSDSK VOLCOPY

ディスクユーティリティは，FM-7システムディスクに記録されています．

4.4.2 バックアップディスクの作成方法

バックアップディスクを作成するために，以下のものを準備します．

・FM-7 システムディスク
・オリジナルディスク

FM-7 システムディスクのバックアップディスクを作成する際は，同一のものとなります．

FM-7 システムディスク

または，

アプリケーションプログラムの記録されたユーザディスク

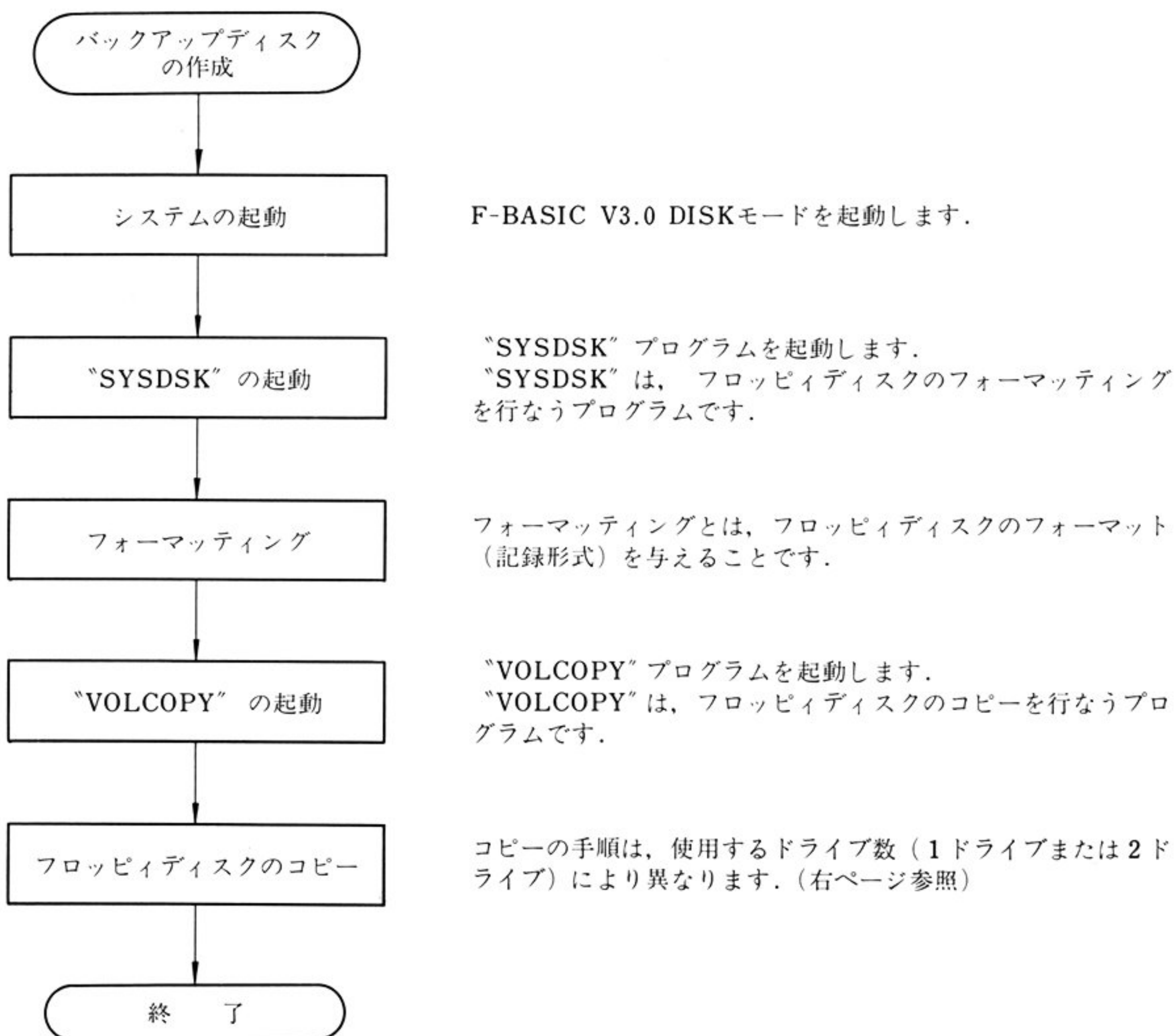
・バックアップディスク

FM-7 システムディスクに添付のフロッピィディスク

または，

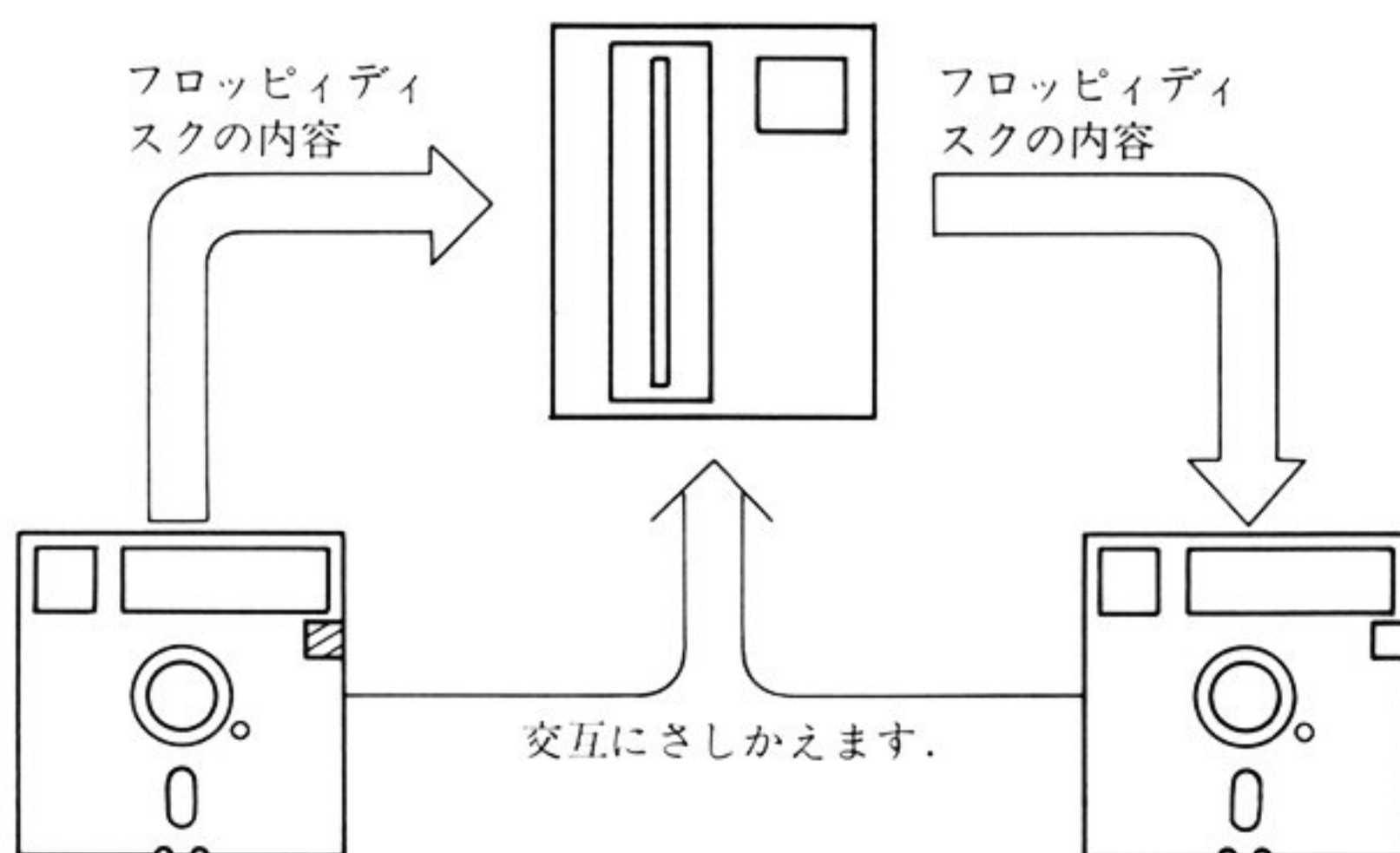
市販のフロッピィディスク

以下に，バックアップディスク作成手順のフローチャートを示します．

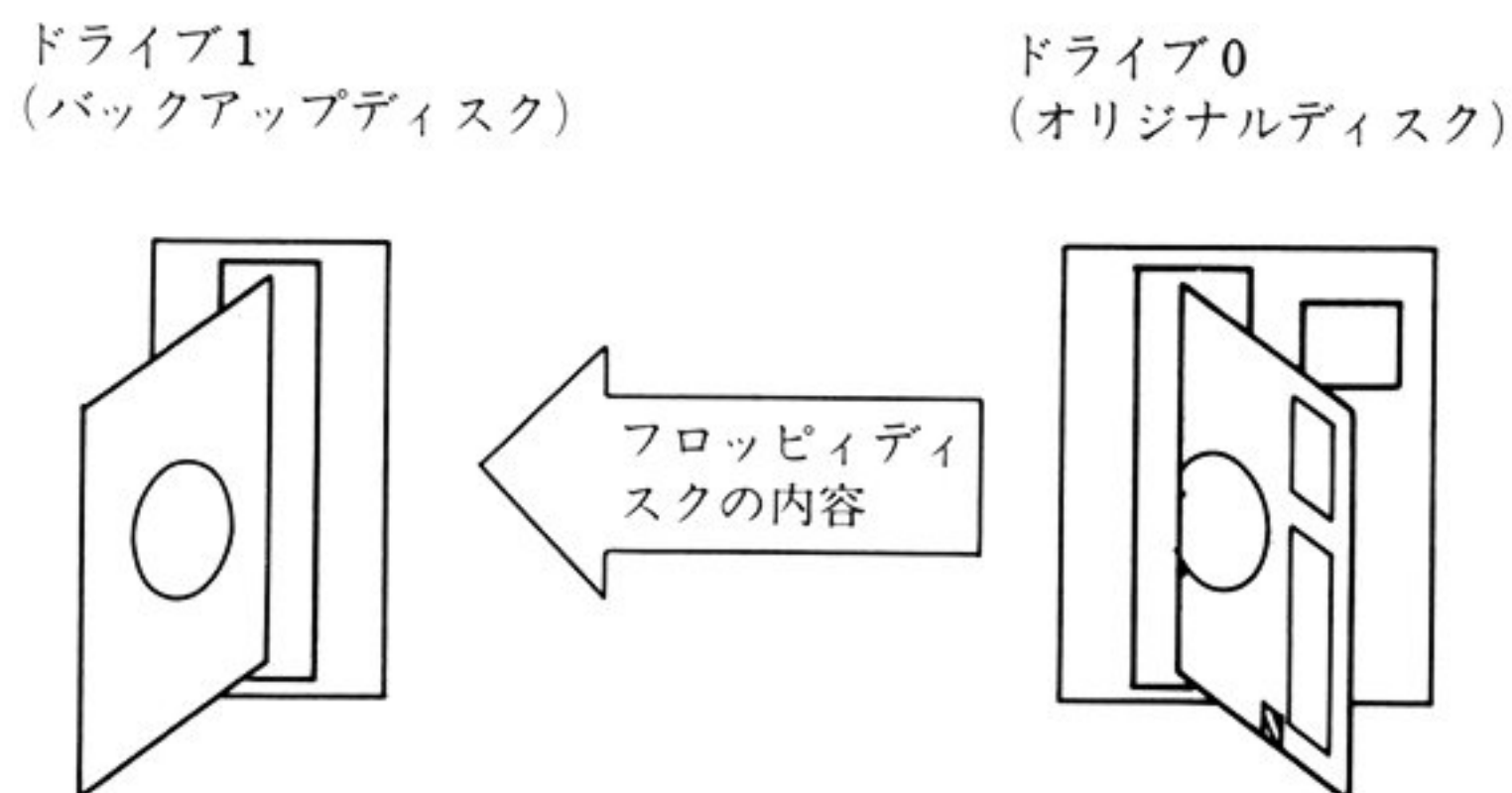
バックアップディスクの作成
↓
システムの起動 — F-BASIC V3.0 DISKモードを起動します．
↓
"SYSDSK" の起動 — "SYSDSK" プログラムを起動します．"SYSDSK" は，フロッピィディスクのフォーマッティングを行なうプログラムです．
↓
フォーマッティング — フォーマッティングとは，フロッピィディスクのフォーマット（記録形式）を与えることです．
↓
"VOLCOPY" の起動 — "VOLCOPY" プログラムを起動します．"VOLCOPY" は，フロッピィディスクのコピーを行なうプログラムです．
↓
フロッピィディスクのコピー — コピーの手順は，使用するドライブ数（1ドライブまたは2ドライブ）により異なります．（右ページ参照）
↓
終　了

フロッピィディスクのコピーを行なう方法は以下の2通りです.

(1) 1ドライブを使用する方法

オリジナルディスクとバックアップディスクをさしかえながら，内容をコピーします.

フロッピィディスクの内容
フロッピィディスクの内容
交互にさしかえます.

(2) 2ドライブを使用する方法

ドライブ0へオリジナルディスク，ドライブ1へバックアップディスクとなるフロッピィディスクをセットし，ドライブ0からドライブ1へコピーします.

ドライブ1
(バックアップディスク)
ドライブ0
(オリジナルディスク)
フロッピィディスクの内容

4.4.3 バックアップディスクの作成手順

(1) 1ドライブを使用する場合

① システムの起動

```
DISK VERSION
How many disk drives      ?
How many disk files(0-15)?

FUJITSU F-BASIC Version 3.0
Copyright (C) 1981 By FUJITSU/MICROSOFT
25775 Bytes Free

Ready
█
```

フロッピィディスクユニットに，FM-7 システムディスクをセットしてF-BASIC V 3.0 DISKモードを起動します．

② "SYSDSK" の起動

```
Ready
RUN"SYSDSK"
```

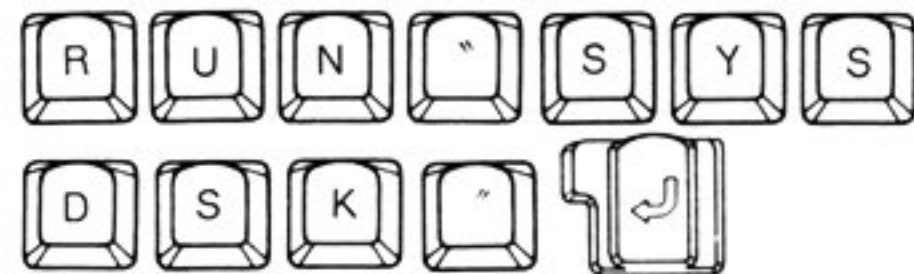
[R][U][N]["][S][Y][S][D][S][K]["][↵]

"SYSDSK" プログラムの起動を行ないます．起動後，メッセージにより条件を入力します．

③ フォーマッティング

```
*** FLOPPY DISK FORMATTING ***

DISK FORMATTING ? Y
WHAT DISK DRIVE ? 0
SET THE FORMATTING DISK ON DRIVE 0
READY ? Y
```

・DISK FORMATTING?

(フォーマッティングを行ないますか？)

[Y][↵] (YESの意味です)

[Y]以外の入力は，次の処理（次頁）へ移ります．

・WHAT DISK DRIVE ?

(バックアップディスクをセットするドライブ番号は？)

[0][↵] （1ドライブの場合は0を入力します.）

・SET THE FORMATTING DISK ON DRIVE 0
READY ?

（フォーマッティングを行なうフロッピィディスクをセットして下さい．準備はできましたか？）

バックアップディスクをディスクユニットへセットした後

[Y] [↵] を入力します．

入力後，フォーマッティングを開始します．

```
DISK FORMATTING COMPLETED !!
*** DISK BASIC SYSTEM COPY ***

1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=  2
READY ? Y
```

〈DISKコード〉
F-BASIC V3.0のディスクに対する命令が記録されています．

フォーマッティング終了後，メッセージを出力し，DISK コードのコピーを行ないます．

・DISK FORMATTING COMPLETED !!

（フォーマッティングの終了です）

・*** DISK BASIC SYSTEM COPY ***
1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=

1ドライブを使用した場合のコピーを行ないます．

2ドライブを使用して，ドライブ0からドライブ1へのコピーを行ないます．

[2]（1ドライブの場合は2を指定します．）

・READY ?（準備はできましたか？）

再度，FM-7 システムディスクをセットした後，[Y] [↵] と入力します．

DISK コードを読込んだ後，メッセージを出力します．

```
SET THE DESTINATION DISK
READY ? Y
```

・SET THE DESTION DISK

（バックアップディスクをセットして下さい.）

FM-7 システムディスクとバックアップディスクをさしかえます.

・READY ?（準備はできましたか？）

さしかえた後

Y ↵ を入力します.

入力後, DISK コードを記録します.

```
SYSTEM CODE COPY COMPLETED !!

Ready
█
```

DISK コードのコピー終了後メッセージを出力し, コマンド受付状態となります.

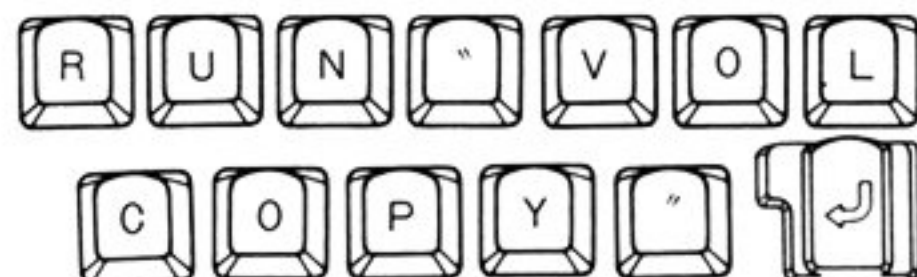
・SYSTEM CODE COPY COMPLETED !!

（DISKコードのコピーを終了しました）

④ "VOLCOPY" の起動

```
Ready
RUN"VOLCOPY"
```

フロッピィディスクを, FM-7 システムディスクとさしかえ, "VOLCOPY" プログラムを起動します.

R U N " V O L C O P Y " ↵

⑤ フロッピィディスクのコピー

```
*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
= 3
READY ? Y
```

"VOLCOPY" プログラムの起動後，メッセセージが出力され，条件を設定します．

```
・*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
=
```

- 3. DRIVE 0 --> 0：1ドライブ使用してコピーを行ないます．
- 2. DRIVE 1 --> 0：2ドライブ使用して，1番から0番へのコピーを行ないます．
- 1. DRIVE 0 --> 1：ドライブを2台使用して0番から1番へのコピーを行ないます．

[3]（1ドライブのコピーは3を指定します．）

・READY ? （準備はできましたか？）

ドライブ0に，オリジナルディスクをセットした後，[Y] [⏎] を入力します．

```
TRACK = 0 1
SET THE DESTINATION DISK !!
READY ? Y
```

・TRACK = 0 1

トラック0と1の2トラックをコピーすることを示します．本体内メモリへトラック0，1の内容が読込まれます．

・SET THE DESTINATION DISK

（バックアップディスクをセットして下さい．）

オリジナルディスクと，バックアップディスクをさしかえます．

・READY ?

（準備はできましたか？）

さしかえた後

[Y] [⏎] を入力します．

入力後，メモリ内に記憶したトラック0，1の内容を記録します．

```
SET THE SOURCE DISK !!
READY ? Y
```

・SET THE SOURCE DISK !!

(オリジナルディスクをセットして下さい.)

オリジナルディスクとバックアップディスクをさしかえます.

・READY ?

(準備はできましたか?)

さしかえた後 [Y] [↵] を入力します.

入力後, 次のトラック2, 3の内容を読込みます.

```
TRACK =  2  3
SET THE DESTINATION DISK !!
READY ? Y
```

トラック2, 3のコピーを行なうメッセージと, 再度フロッピィディスクをさしかえるよう, メッセージが出力されます.

さしかえた後 [Y] [↵] を入力します.

以下同様に, ディスクのさしかえと, キー入力を行ないます.

・"SET THE DESTINATION DISK"のメッセージにより, バックアップディスクをセットしまます.

・"SET THE SOURCE DISK"のメッセージによりオリジナルディスクをセットします.

・各フロッピィディスクをセットした後, [Y] [↵] を入力します.

・以上の手順を40トラック分繰返します.

```
READY ? Y
TRACK =  38  39
SET THE DESTINATION DISK !!
READY ? Y
COPY FINISHED!

Ready
■
```

トラック38，39をコピーした後コピー終了のメッセージを出力し，コマンド受付状態となります．

・COPY FINISHED !!

（コピーは終了しました）

⑥ 内容の確認

```
Ready
FILES

DFMCD      2 B S 1
MCOPY      2 B S 1
SYSDSK     0 B S 1
VOLCOPY    0 B S 1
AUTOUTY    0 B S 1
PFDEF      0 B S 2
DEMO1      0 B S 7
DEMO2      0 B S 1
DEMO21     0 B S 8
DEMOSUB    2 B S 3

126 Clusters Free

Ready
■
```

FILES 命令により，フロッピィディスクの内容を確認します．

F I L E S ↵

⑦ 以下に今までの手順を示します.

```
DISK VERSION
How many disk drives      ?
How many disk files(0-15)?

FUJITSU F-BASIC Version 3.0
Copyright (C) 1981 By FUJITSU/MICROSOFT
25775 Bytes Free

Ready
RUN"SYSDSK"

*** FLOPPY DISK FORMATTING ***

DISK FORMATTING ? Y
WHAT DISK DRIVE ? 0
SET THE FORMATTING DISK ON DRIVE 0
READY ? Y
DISK FORMATTING COMPLETED !!
*** DISK BASIC SYSTEM COPY ***

1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=  2
READY ? Y
SET THE DESTINATION DISK
READY ? Y
SYSTEM CODE COPY COMPLETED !!

Ready
RUN"VOLCOPY"
*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
=  3
READY ? Y
TRACK =  0  1
SET THE DESTINATION DISK !!
READY ? Y
SET THE SOURCE DISK !!
READY ? Y
TRACK =  2  3
SET THE DESTINATION DISK !!
READY ? Y
        .
        .
        .
        .
TRACK =  38  39
SET THE DESTINATION DISK !!
READY ? Y
COPY FINISHED!

Ready
█
```

(2) 2ドライブを使用する場合

① システムの起動

```
DISK VERSION
How many disk drives      ?
How many disk files(0-15)?

FUJITSU F-BASIC Version 3.0
Copyright (C) 1981 By FUJITSU/MICROSOFT
25775 Bytes Free

Ready
■
```

フロッピィディスクユニットに，FM-7 システムディスクをセットして F-BASIC V3.0 DISKモードを起動します．

② "SYSDSK" の起動

```
Ready
RUN"SYSDSK"
```

R U N " S Y S D S K " ↵

"SYSDSK" プログラムの起動を行ないます．起動後，メッセージにより条件を入力します．

③ フォーマッティング

```
*** FLOPPY DISK FORMATTING ***

DISK FORMATTING ? Y
WHAT DISK DRIVE ? 1
SET THE FORMATTING DISK ON DRIVE 1
READY ? Y
```

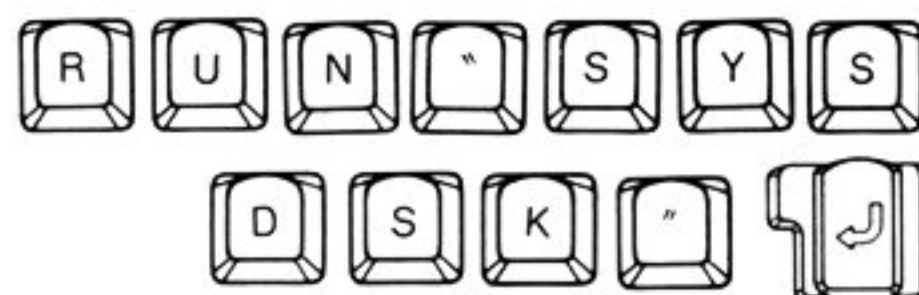
・DISK FORMATTING ?

(フォーマッティングを行ないますか？)

Y ↵ (YESの意味です)

Y 以外の入力は，次の処理（次頁）へ移ります．

・WHAT DISK DRIVE ?

(バックアップディスクをセットするドライブ番号は？)

1 ↵ (ドライブ1番へセットしますので1を入力します.)

```
DISK FORMATTING COMPLETED !!
*** DISK BASIC SYSTEM COPY ***

1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=  1
READY ? Y
```

<DISKコード>
F-BASIC V3.0のディスクに対する命令が記録されています.

・SET THE FORMATTING DISK ON DRIVE 1 READY ?

（フォーマッティングを行なうフロッピィディスクをセットして下さい. 準備はできましたか？）

バックアップディスクをドライブ1番へセットした後

[Y] [↵] を入力します.
入力後, フォーマッティングを開始します.

フォーマッティング終了後, メッセージを出力し, DISKコードのコピーを行ないます.

・DISK FORMATTING COMPLETED !!

（フォーマッティングの終了です）

```
・*** DISK BASIC SYSTEM COPY ***
  1. DRIVE 0 -->1
  2. DRIVE 0 -->0
  =
```

1ドライブを使用した場合のコピーを行ないます.

2ドライブを使用して, ドライブ0からドライブ1へのコピーを行ないます.

[1] （2ドライブ使用する場合は1を指定します.）

・READY ?（準備はできましたか？）

ドライブ0番へFM-7システムディスク, ドライブ1番へバックアップディスクがセットされているのを確認した後

[Y] [↵] と入力します.
入力後, DISKコードのコピーを行ないます.

```
SYSTEM CODE COPY COMPLETED !!

Ready
█
```

DISK コードのコピー終了後メッセージを出力し，コマンド受付状態となります．

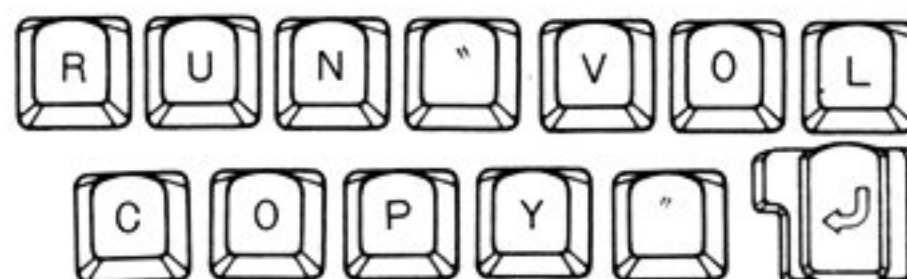
・SYSTEM CODE COPY COMPLETED !!

（DISK コードのコピーを終了しました）

④ "VOLCOPY" の起動

```
Ready
RUN"VOLCOPY"
```

FM-7 システムディスクの "VOLCOPY" プログラムを起動します．

R U N " V O L C O P Y " ↵

⑤ フロッピィディスクのコピー

```
*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
=  1
READY ? Y
```

"VOLCOPY" プログラムの起動後，メッセージが出力され，条件を設定します．

・*** DISK COPY ***
1. DRIVE 0 -->1
2. DRIVE 1 -->0
3. DRIVE 0 -->0

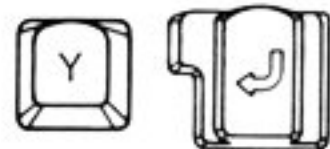
3. DRIVE 0 -->0	1 ドライブ使用してコピーを行ないます．
2. DRIVE 1 -->0	2 ドライブ使用して，1 番から 0 番へのコピーを行ないます．
1. DRIVE 0 -->1	2 ドライブを使用して 0 番から 1 番へのコピーを行ないます．

1 （2 ドライブのコピーは 1 を指定します．）

```
TRACK =  0  1
TRACK =  2  3
TRACK =  4  5
TRACK =  6  7
      .
      .
      .
      .
      .
```

・READY ?

(準備はできましたか?)

ドライブ0へオリジナルディスク, ドライブ1へバックアップディスクをセットした後, [Y] [↵] を入力します.

入力後, コピーを開始します.

コピーは, トラック番号を表示しながら行ないます.

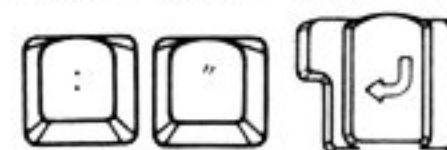
```
      .
      .
      .
      .
      .
TRACK =  34  35
TRACK =  36  37
TRACK =  38  39
COPY FINISHED!

Ready
█
```

トラック39までコピーを行なうと, コピー終了のメッセージを出力して, コマンド受付状態となります.

・COPY FINISHED !

(コピーは終了しました)

⑥ 内容の確認

```
Ready
FILES"1:"

DFMCD      2 B S 1
MCOPY      2 B S 1
SYSDSK     0 B S 1
VOLCOPY    0 B S 1
AUTOUTY    0 B S 1
PFDEF      0 B S 2
DEMO1      0 B S 7
DEMO2      0 B S 1
DEMO21     0 B S 8
DEMOSUB    2 B S 3

126 Clusters Free

Ready
█
```

[F] [I] [L] [E] [S] ["] [1] [:] ["] [↵]

FILES 命令により, フロッピィディスクの内容を確認します.

⑦ 手順例

以下に，今までの手順を示します．

```
DISK VERSION
How many disk drives     ?
How many disk files(0-15)?

FUJITSU F-BASIC Version 3.0
Copyright (C) 1981 By FUJITSU/MICROSOFT
25775 Bytes Free

Ready
RUN"SYSDSK"

*** FLOPPY DISK FORMATTING ***

DISK FORMATTING ? Y
WHAT DISK DRIVE ? 1
SET THE FORMATTING DISK ON DRIVE 1
READY ? Y
DISK FORMATTING COMPLETED !!
*** DISK BASIC SYSTEM COPY ***

1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=  1
READY ? Y
SYSTEM CODE COPY COMPLETED !!

Ready
RUN"VOLCOPY"
*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
=  1
READY ? Y
TRACK =  0  1
TRACK =  2  3
TRACK =  4  5
TRACK =  6  7
      .
      .
      .
      .
      .
TRACK =  34  35
TRACK =  36  37
TRACK =  38  39
COPY FINISHED!

Ready
█
```

4.4.4 バックアップディスク作成上の注意

〝VOLCOPY〞プログラム実行中，エラー（COPY PROGRAM ABORTING）が発生した場合は，再度バックアップディスクのフォーマッティングを行なって下さい．

フォーマッティングを行なったにもかかわらず，エラーが発生したときは，以下の原因が考えられますので，他のフロッピィディスクを御使用下さい．

〈エラー発生原因〉

1．フロッピィディスクのキズ

2．フロッピィディスクの不良

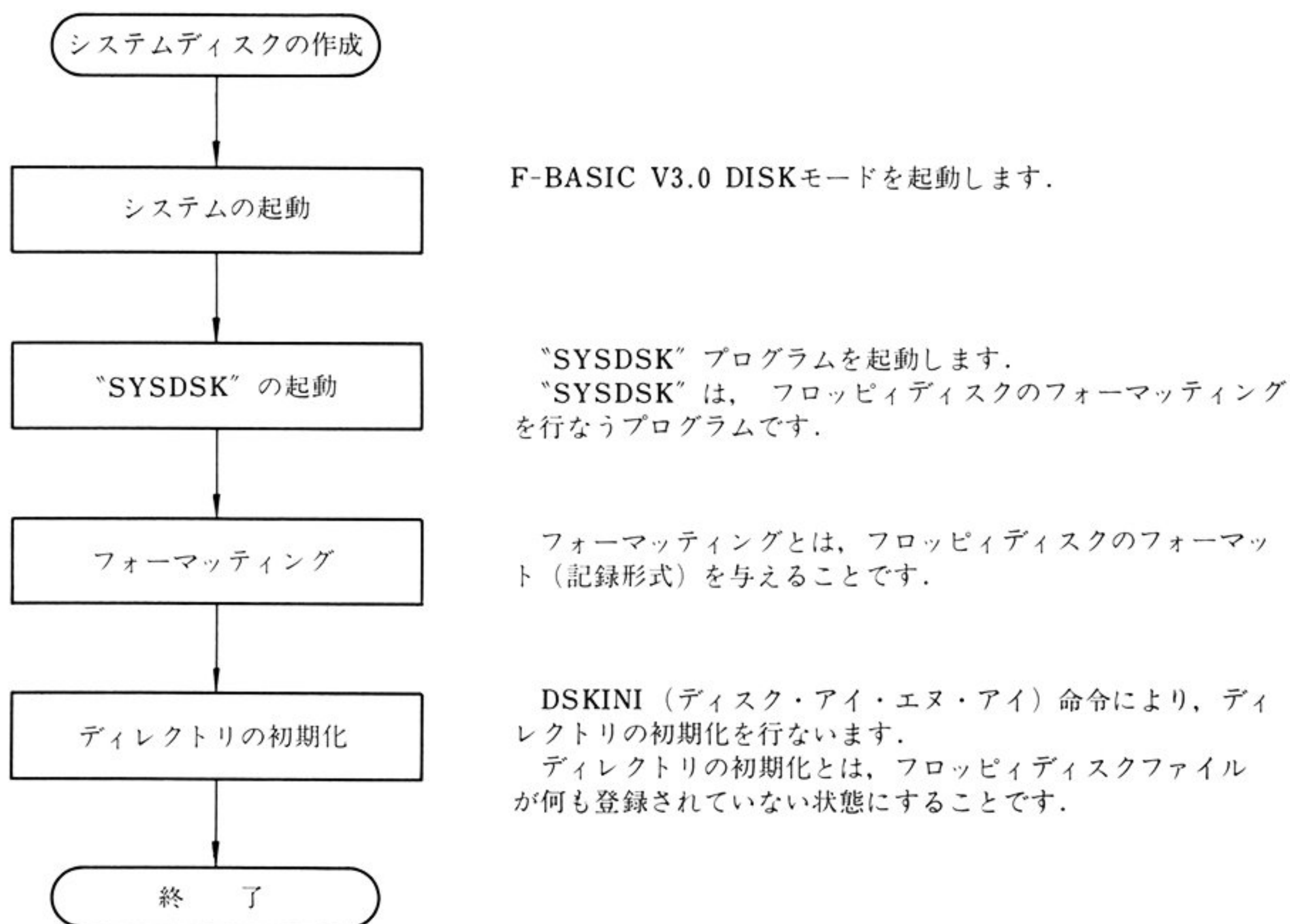
4.4.5 システムディスクの作成方法

システムディスクを作成するために，以下のものを準備します．

・FM-7 システムディスク

・システムディスクとなるフロッピィディスク

以下に，システムディスク作成手順のフローチャートを示します．

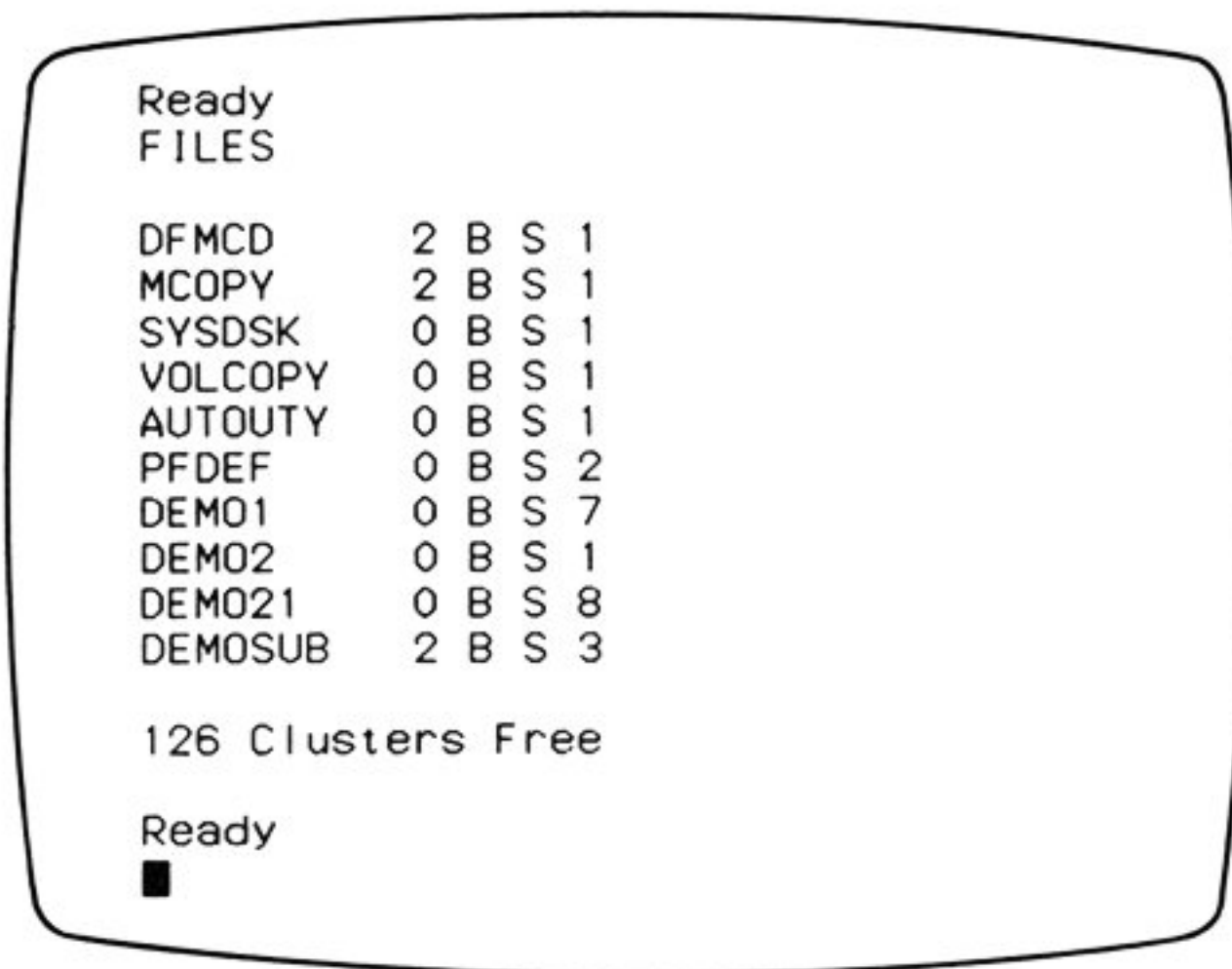
(システムディスクの作成)
↓
[システムの起動] — F-BASIC V3.0 DISKモードを起動します．
↓
["SYSDSK" の起動] — "SYSDSK" プログラムを起動します．"SYSDSK" は，フロッピィディスクのフォーマッティングを行なうプログラムです．
↓
[フォーマッティング] — フォーマッティングとは，フロッピィディスクのフォーマット（記録形式）を与えることです．
↓
[ディレクトリの初期化] — DSKINI（ディスク・アイ・エヌ・アイ）命令により，ディレクトリの初期化を行ないます．ディレクトリの初期化とは，フロッピィディスクファイルが何も登録されていない状態にすることです．
↓
(終　了)

システムディスクの作成手順は，『13−45ページ　13.11.1　システムディスクの作成』を参照して下さい．

4.4.6 FM-7システムディスクの内容

バックアップディスクやシステムディスクを作成するためには，FM-7 システムディスクに記録されている，ディスクユーティリティを使用します．

以下に，FM-7 システムディスクに記録されているプログラムファイルを示します．

ファイル名の出力画面

```
Ready
FILES

DFMCD      2 B S 1
MCOPY      2 B S 1
SYSDSK     0 B S 1
VOLCOPY    0 B S 1
AUTOUTY    0 B S 1
PFDEF      0 B S 2
DEMO1      0 B S 7
DEMO2      0 B S 1
DEMO21     0 B S 8
DEMOSUB    2 B S 3

126 Clusters Free

Ready
■
```

F I L E S ↵

FILES（ファイルズ）命令は，ミニフロッピィディスクに記録されたファイル名をディスプレイへ出力します．

(1) DFMCD……………SYSDSK補助プログラム

(2) MCOPY……………VOLCOPY補助プログラム

(3) SYSDSK…………システムディスク作成用プログラム

(4) VOLCOPY………バックアップディスク作成用プログラム

(5) AUTOUTY………自動スタートユーティリティ

(6) PFDEF……………PFキーの設定ユーティリティ

(7) DEMO1……………デモプログラム

(8) DEMO2……………デモプログラム

(9) DEMO21…………デモプログラム

(10) DEMOSUB………(7)，(8)により使用される機械語プログラム

(1)～(6)が，ディスクユーティリティであり，使用方法は，13－44ページ13．11ディスクユーティリティリティの項を参照して下さい．

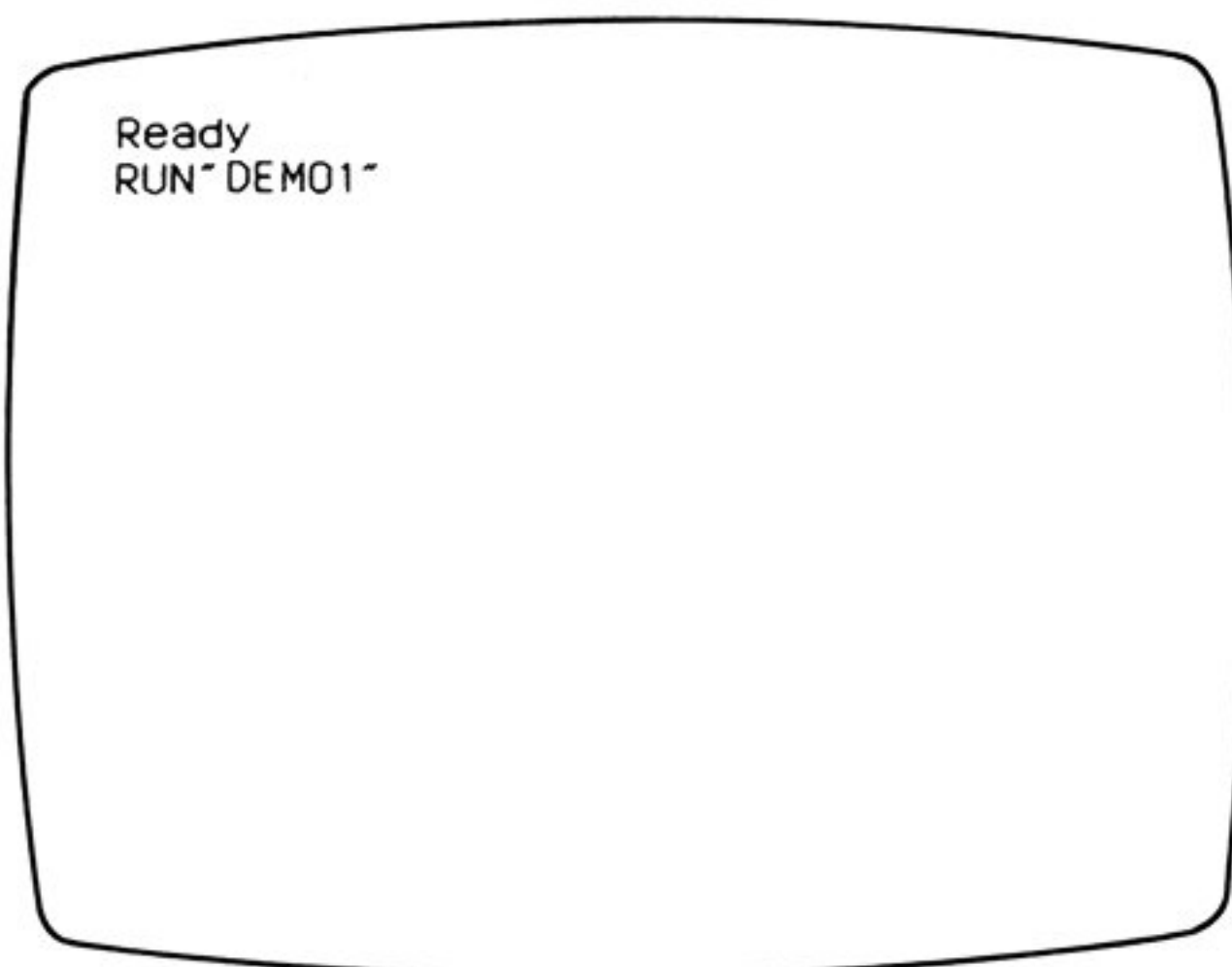
4.4.7 デモンストレーションプログラム (DEMO1, DEMO2)

FM-7 システムディスクには，以下のデモンストレーションプログラムが記録されています．

・DEMO 1　　　　・DEMO21

・DEMO 2　　　　・SUB（DEMO 1およびDEMO 2により使用される機械語プログラムです．）

このプログラムは，連結され，一つのプログラムのようになっていますので，DEMO 1またはDEMO 2のどちらかを起動します．

（1） プログラムの起動

以下のようにキー入力します．

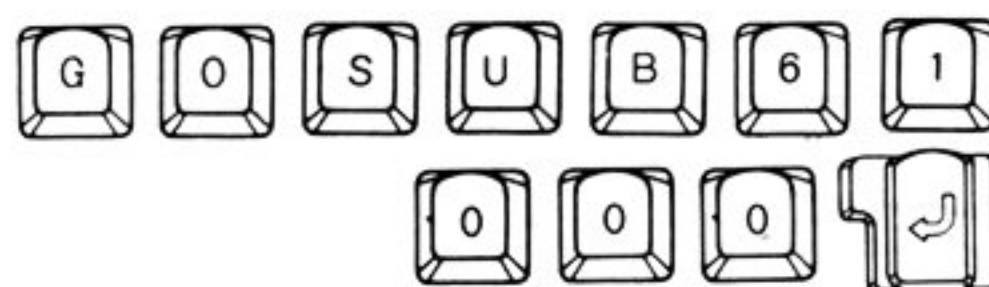
または，

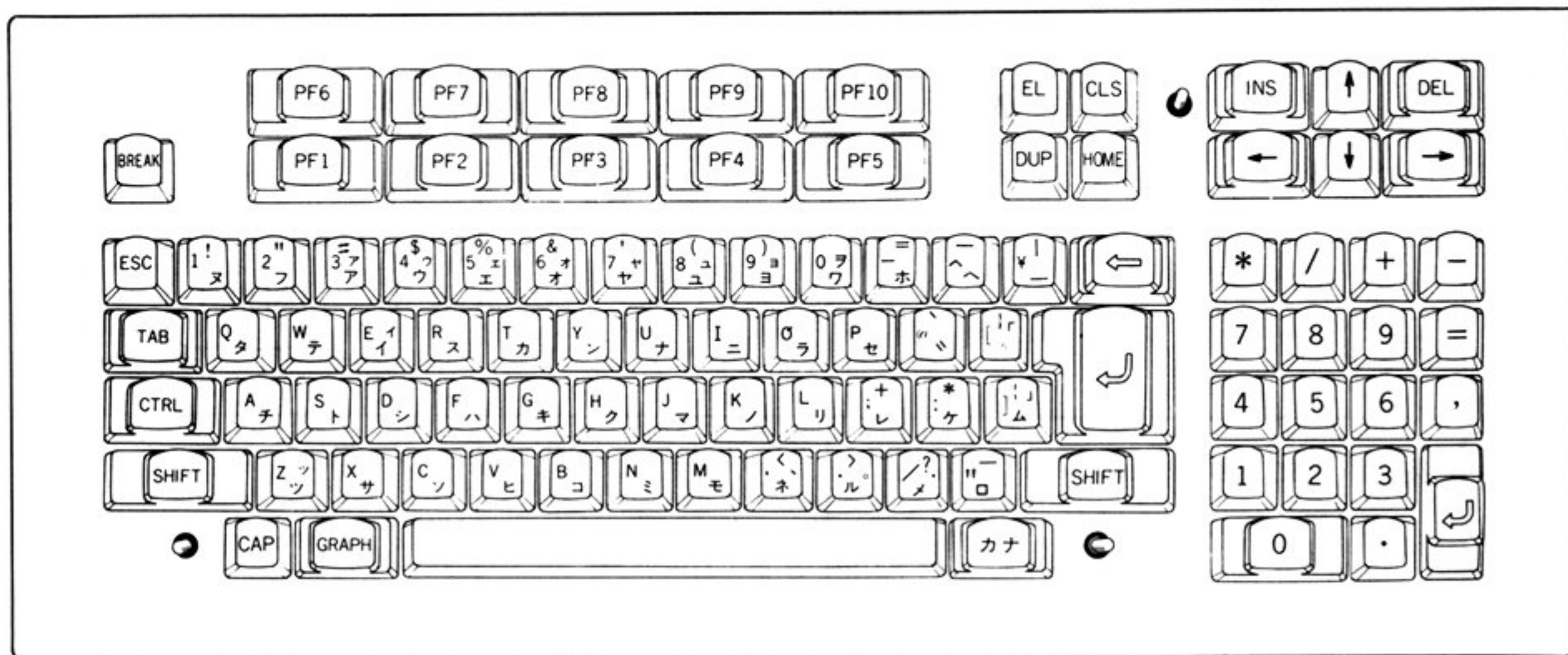
〈実行画面の一部〉

(2) プログラムの停止

```
Break In 61050
Ready
GOSUB61000
```

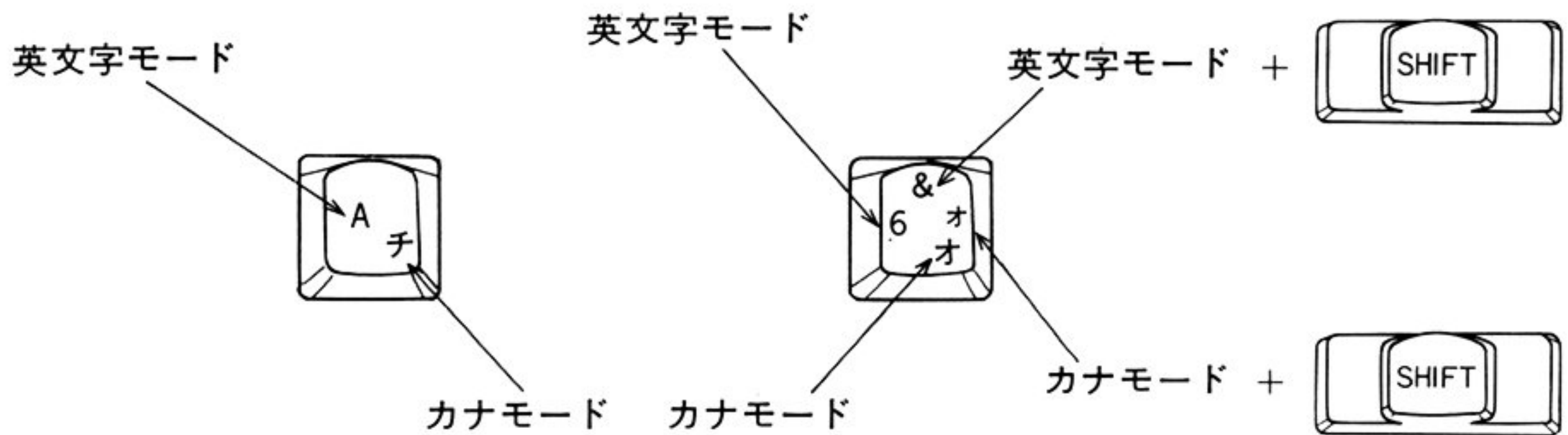
デモンストレーションプログラムは繰返し実行しますので，停止する場合は BREAK を押します．さらに，機械語プログラムの実行を停止するため，以下のようにキー入力を行ないます．

G O S U B 6 1 0 0 0 ↵

第5章 キーボード

FM-7 のキーボードは，JIS 配列に準拠し，専用のマイクロコンピュータ（MB88401）により制御されています．また，シリンドリカル・ステップ・スカルプチャの形状になっており，操作性がすぐれています．

5.1 キーボード概要

FM-7 のキーボードは，英文字モード，カナモードの2つのモードがあり，それぞれに，SHIFT キーを併用したシフト機能により，1つのキーで最大4文字の出力が可能です．

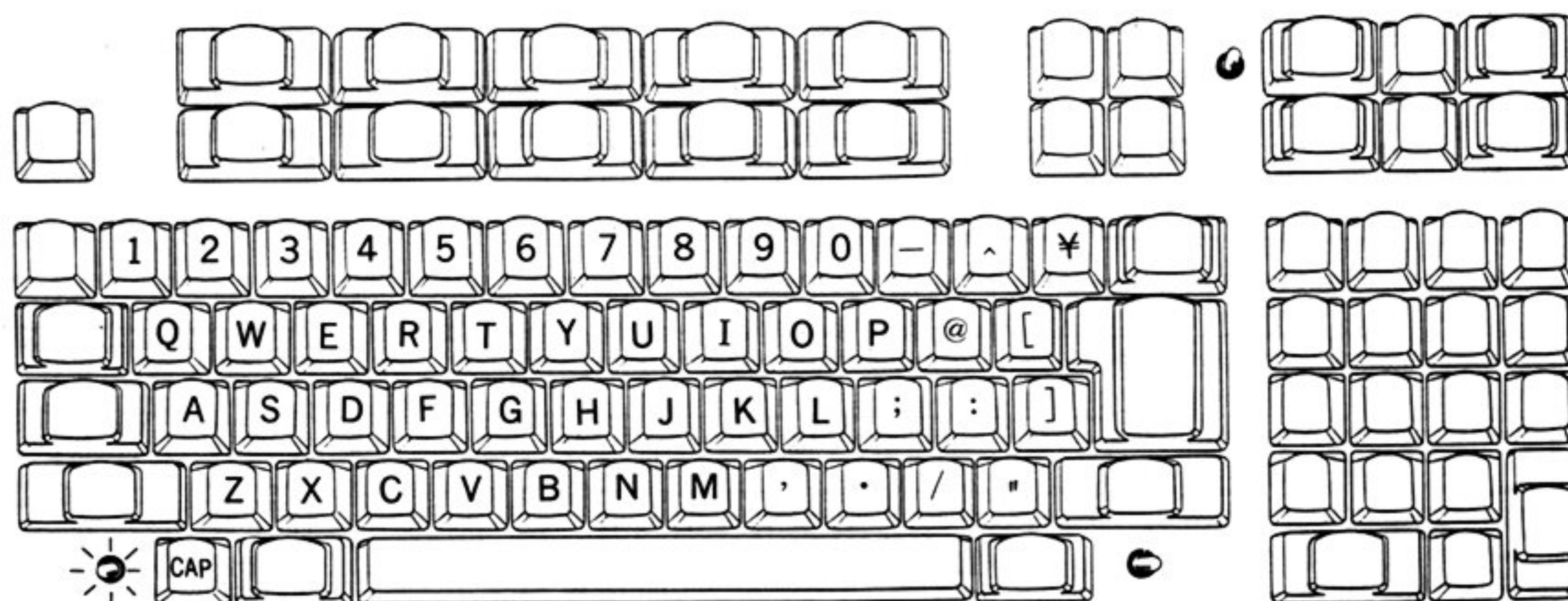
英文字モード
A
チ
カナモード

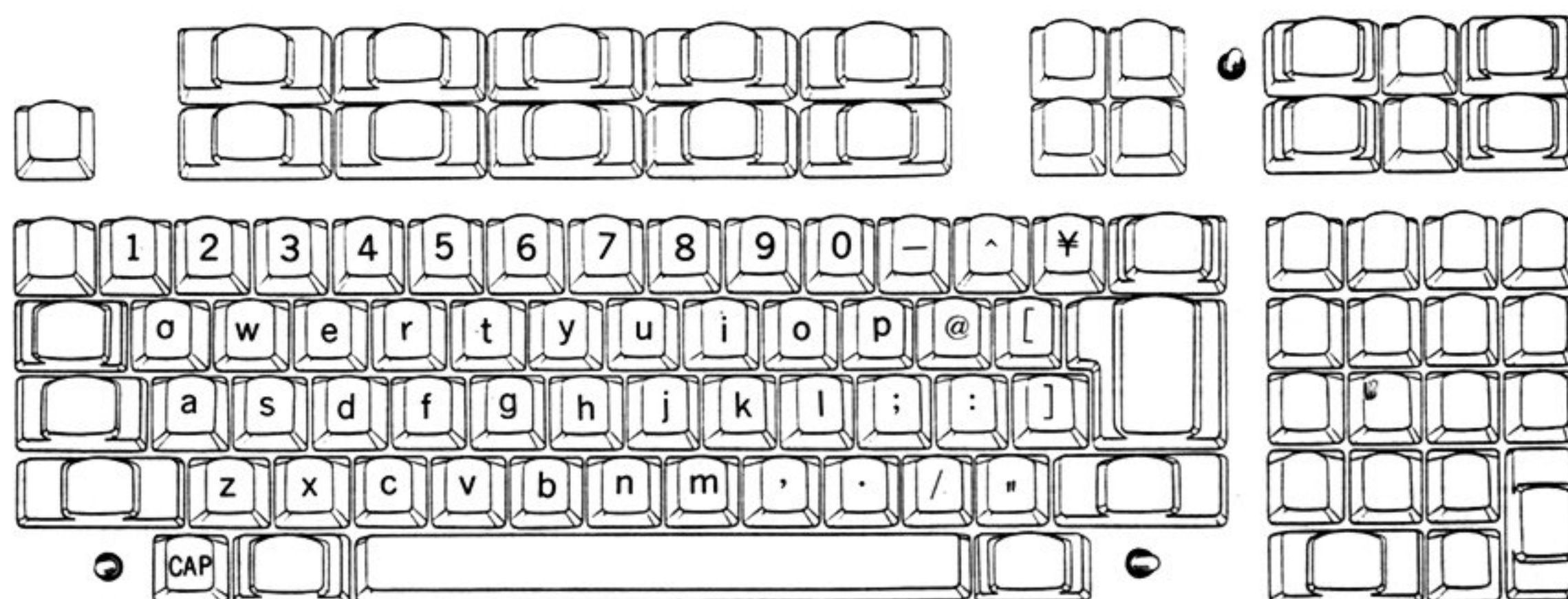
英文字モード
英文字モード ＋ SHIFT
&
6
ォ
オ
カナモード
カナモード ＋ SHIFT

モード		CAP	カ　ナ	SHIFT
英文字モード	英大文字	☀	○	
		○	○	＋ SHIFT
	英小文字	○	○	
		☀	○	＋ SHIFT
カナモード	・カナ ・濁点 ・半濁点 ・記号	☀ または ○	☀	
	・カナ小文字 ・句読点 ・記号 ・わ行のヲ	☀ または ○	☀	＋ SHIFT

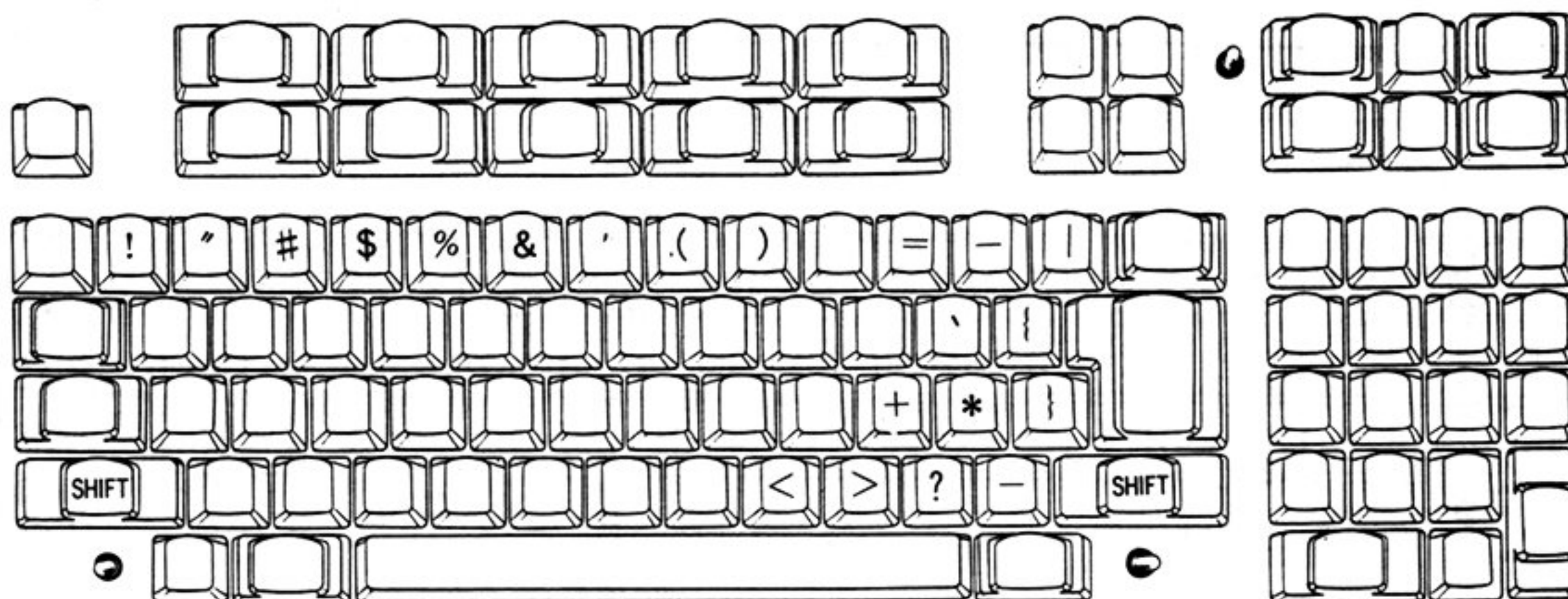
☀：点灯　○：消灯

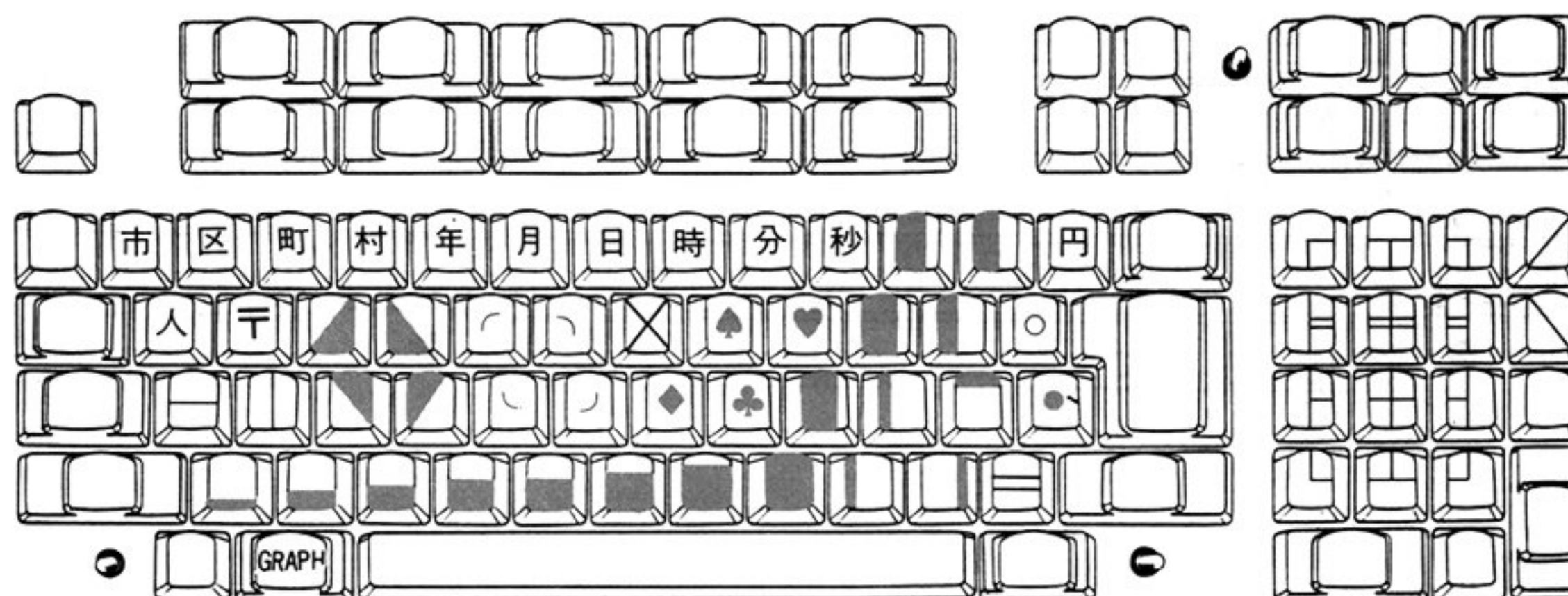
5.1.1 英文字モード

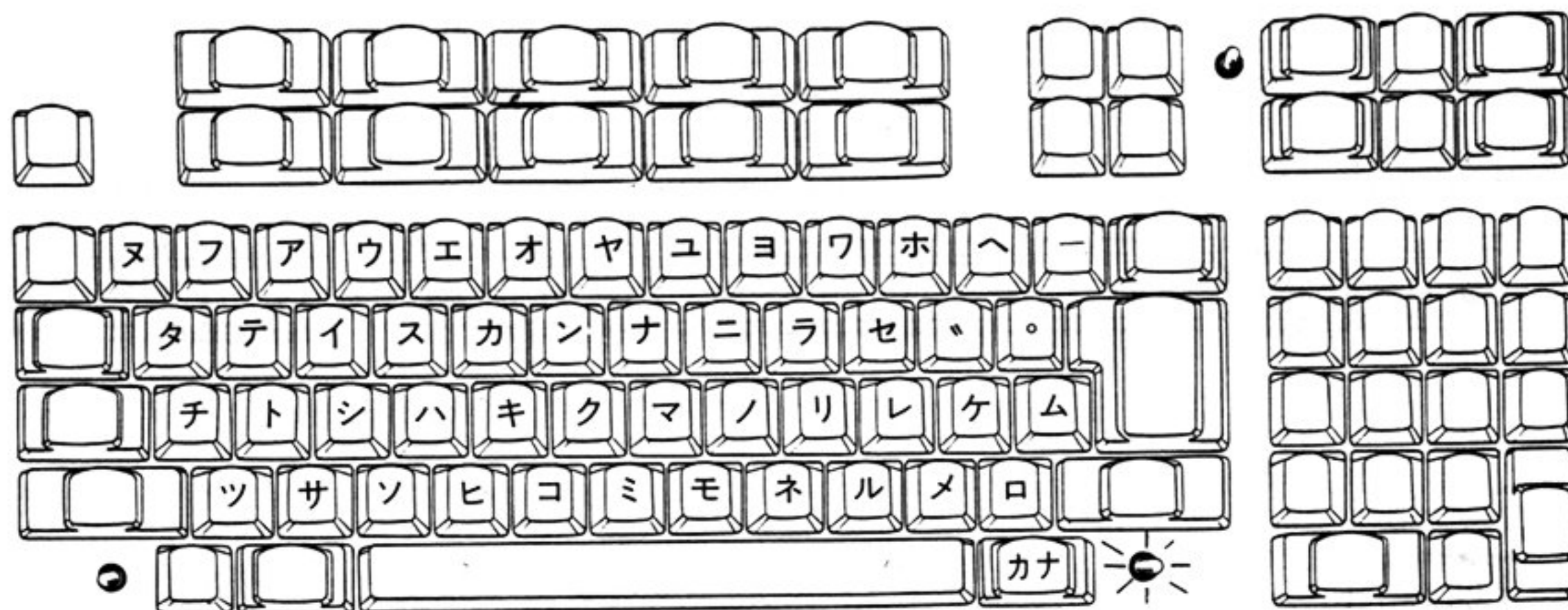
(1) 英大文字モード

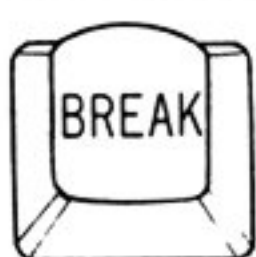
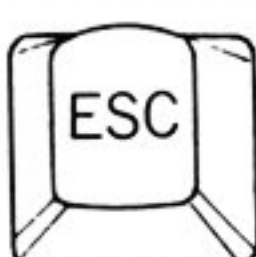
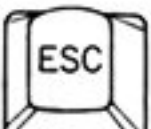
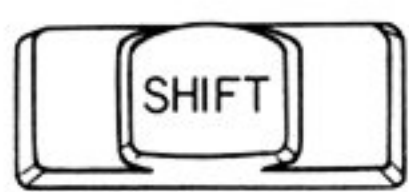
CAP ランプが点灯しているときに英大文字を出力します.

(2) 英小文字モード

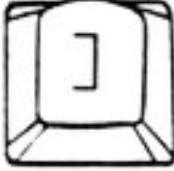
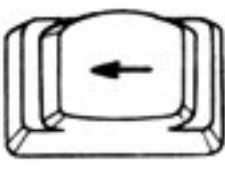
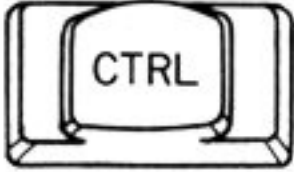
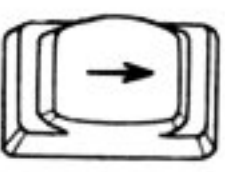
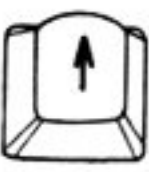
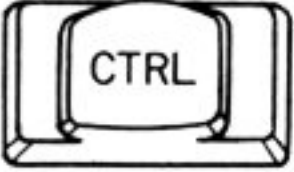
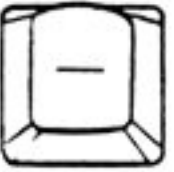
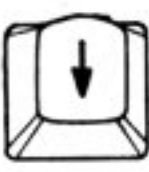
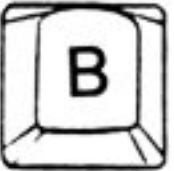
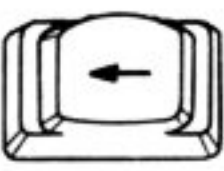
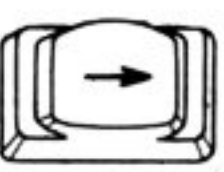
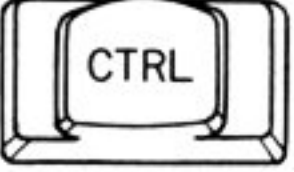
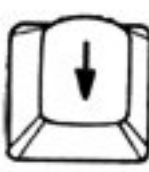
CAP ランプが消灯しているときに英小文字を出力します.

(3) 英文字モードのシフト

英文字モードにして，シフトキーを押しながらキーを押しますと，次の記号が出力されます．シフトキーを押しながらアルファベットを入力しますと，CAPランプが点灯時は小文字，CAPランプが消灯時は大文字が出力されます．

5.1.2 グラフィックモード

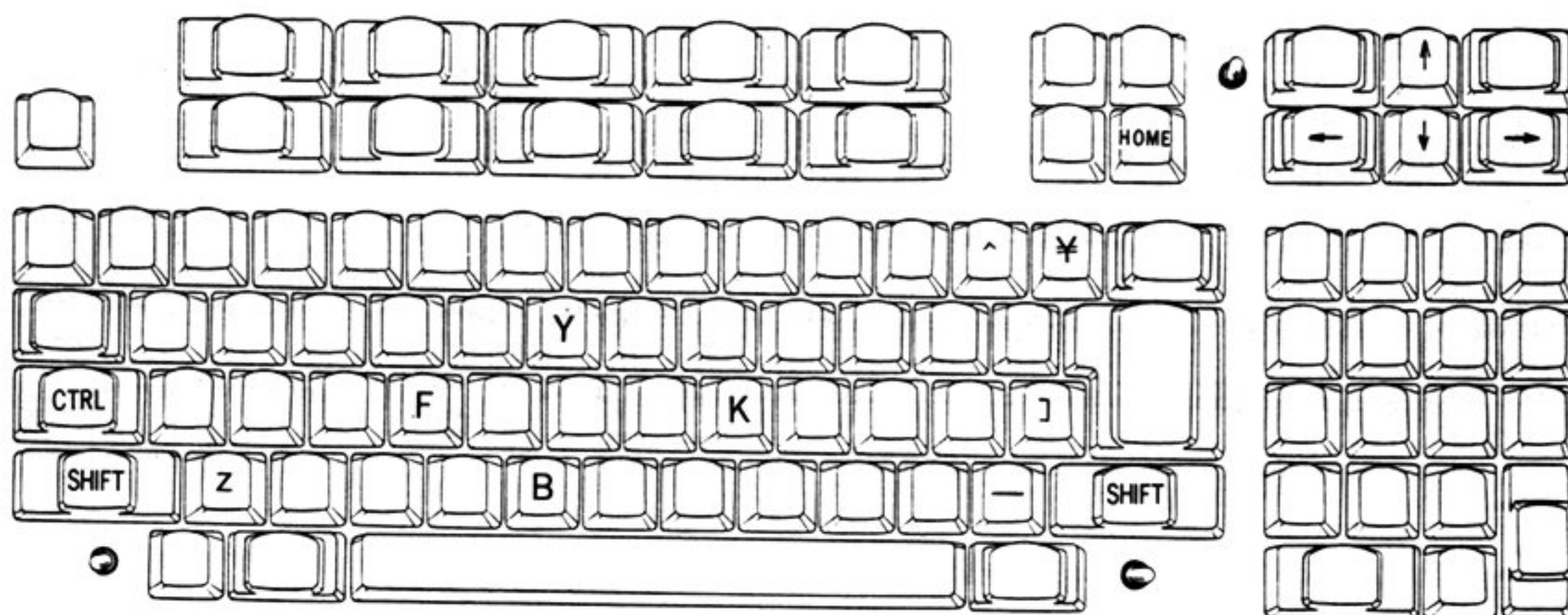
グラフィックキーを押しながらキーを押しますと，次のグラフィックパターンが出力されます．

5.1.3 カナモード

カナキーを押しますと，カナモードとなりカナランプが点灯します．キーの右下のカナ，濁点，半濁点，記号が出力されます．

カナモード シフト

カナモード中にて，シフトキーを押しながら下のキーを押しますと，カナ小文字，句読点，記号，わ行のヲが出力されます．

5.2 特殊キー

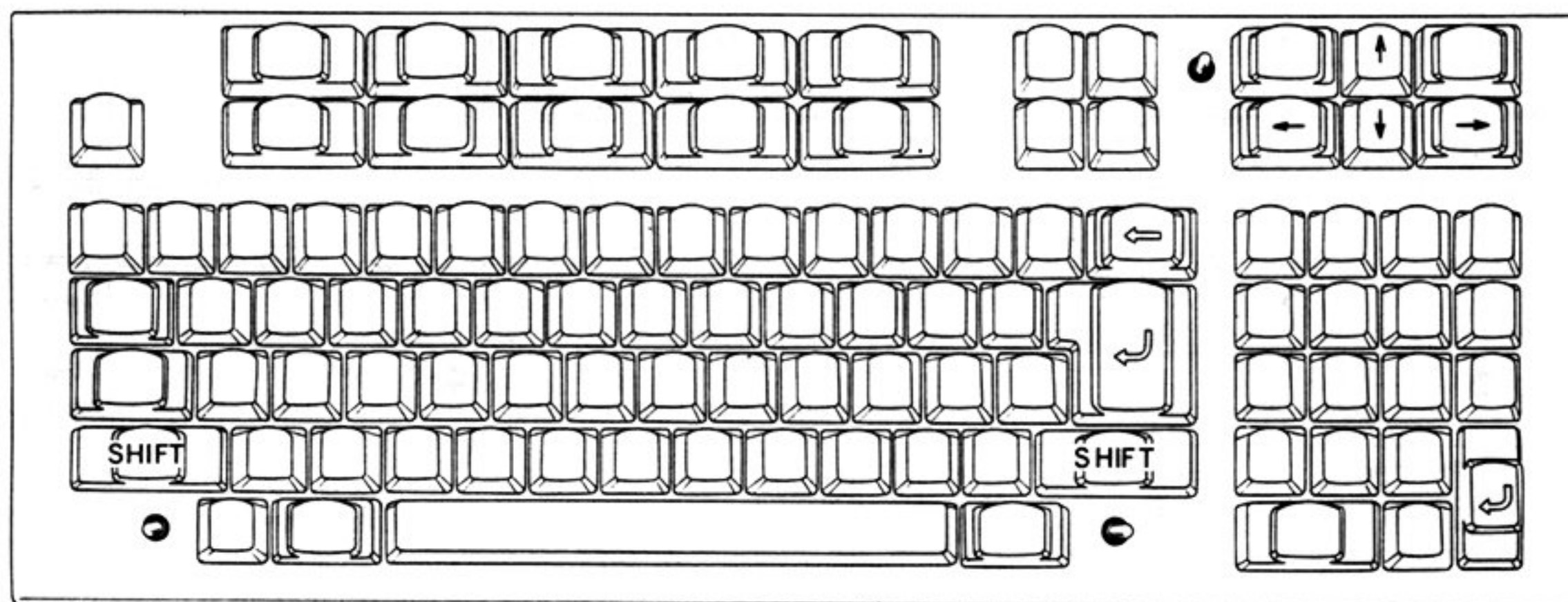
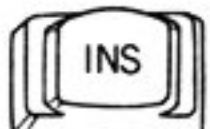
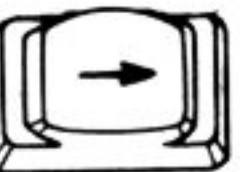
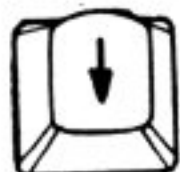
CAP キャピタル キー (capital key)	このキーが押されると，英大文字モードに設定され，CAP キー左側の状態表示ランプが点灯します．このとき，英字キーを押すと，英大文字が表示されます． 英大文字モードの解除は，もう一度 CAP キーを押しますと，状態表示ランプが消え，英小文字モードになります．
グラフィック キー (graphic key) GRAPH	このキーは，文字キーと併用して使用します．5－4ページに表示されているグラフィックパターンが出力されます．
カナ キー カナ	このキーが押されると，カナモードが設定され，カナ キー右側の状態表示が点灯し，カタカナを入力することができます． カナモードの解除は，もう一度 カナ キーを押しますと，状態表示ランプが消え，英文字モードになります．

キー	説明
BREAK ブレイク キー (break key)	このキーが押されるとプログラムの実行が中断されます.
ESC エスケープ キー (escape key)	このキーが押されるとLISTコマンドにより画面表示されている文字の出力を一時停止します. この状態でもう一度 ESC キーを押しますと, 次の行の文字列を表示し, 停止します. 停止状態の解除は BREAK キーを除くいずれかのキーを押します.
TAB タブ キー (tab key)	このキーが入力されると, カーソルは次のタブ位置(8桁単位)へ移動します. このとき, カーソルの通過した部分はスペース(空白)になります.
CTRL コントロール キー (control key)	このキーは他のキーと併用して使います. (p. 5-8参照).
SHIFT シフト キー (shift key)	このキーは他のキーと併用して使います. (p. 5-10参照).

コントロール キー (Control key)

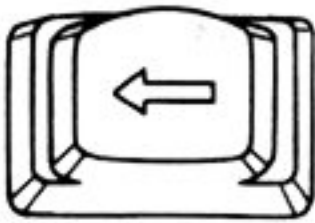
CTRL Y F K] SHIFT Z B — SHIFT HOME ^ ¥

コントロールキーは，他のキーと同時に押下することにより，次のような意味を持ちます．

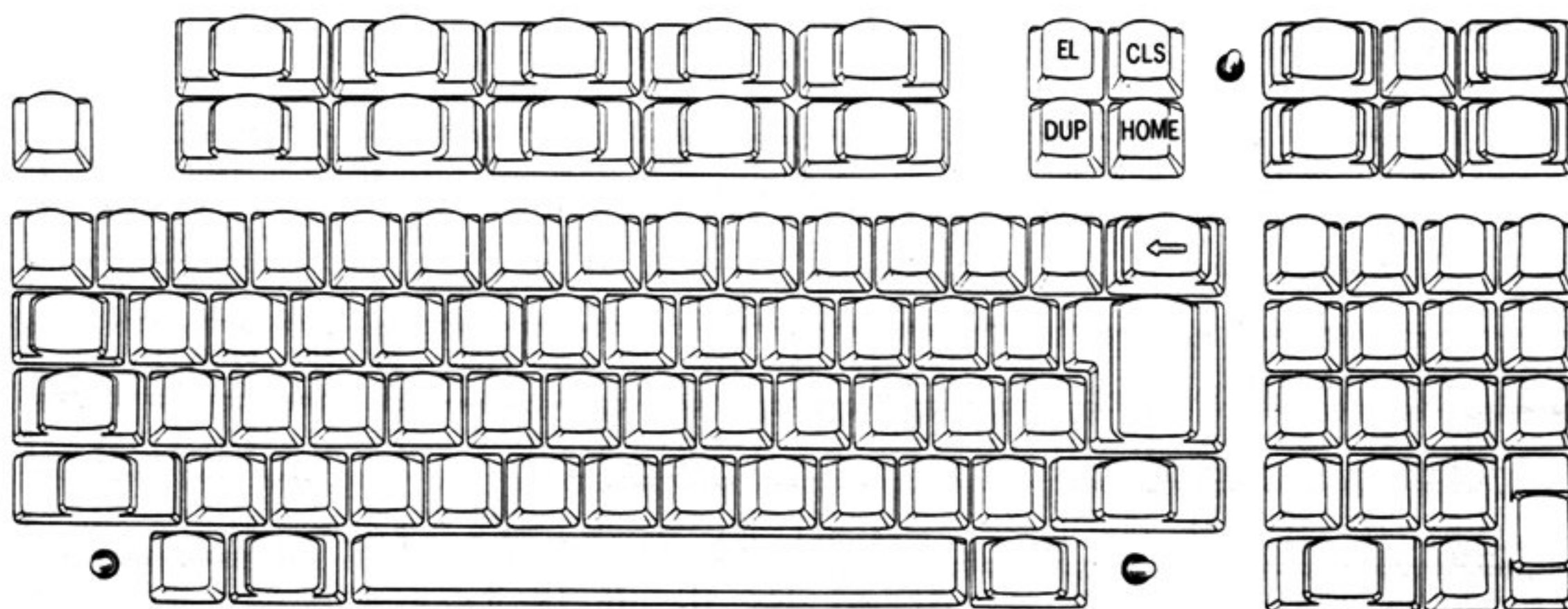
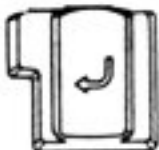
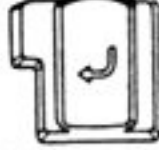
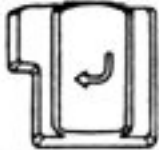
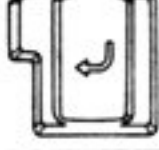
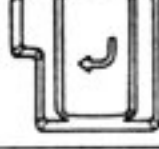
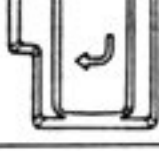
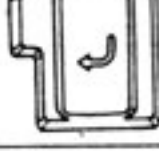
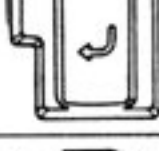
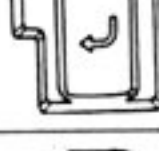
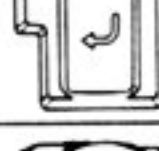
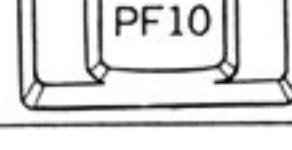
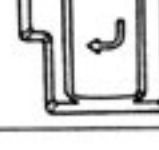
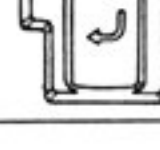
CTRL +]	← キーを押したのと同じ結果が得られます．
CTRL + ¥	→ キーを押したのと同じ結果が得られます．
CTRL + ^	↑ キーを押したのと同じ結果が得られます．
CTRL + —	↓ キーを押したのと同じ結果が得られます．
CTRL + B	SHIFT + ← の動作と同じ動作をします．
CTRL + F	SHIFT + → の動作と同じ動作をします．
CTRL + Y	SHIFT + ↑ の動作と同じ動作をします．
CTRL + Z	SHIFT + ↓ の動作と同じ動作をします．
CTRL + K	HOME キーを押したのと同じ結果が得られます．

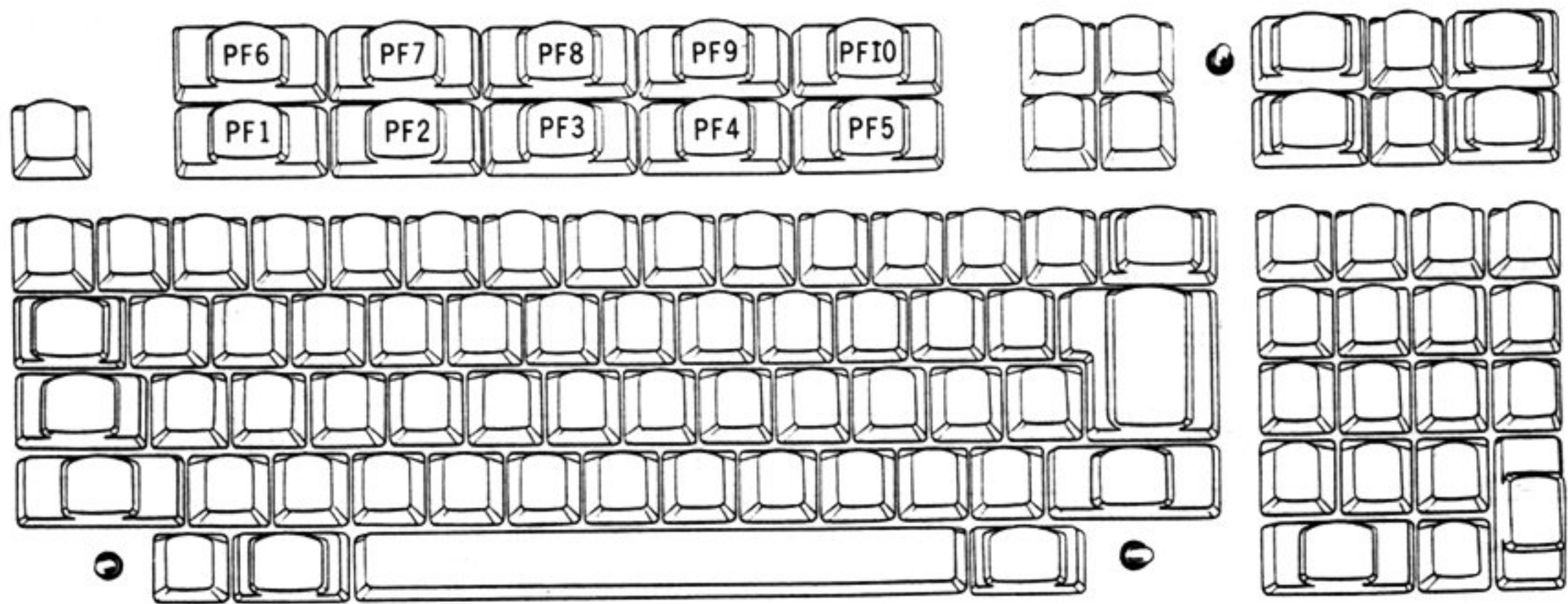
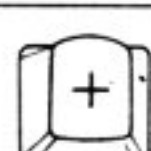
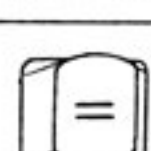
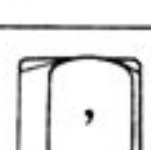
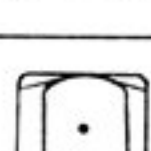
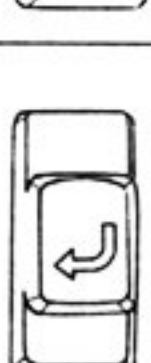
キー操作	対応キー	説明
CTRL + R	INS	キーを押したのと同じ結果が得られます.
CTRL + H	⇦	キーを押したのと同じ結果が得られます.
CTRL + E	EL	キーを押したのと同じ結果が得られます.
CTRL + I	TAB	キーを押したのと同じ結果が得られます.
CTRL + Q	DUP	キーを押したのと同じ結果が得られます.
CTRL + [	ESC	キーを押したのと同じ結果が得られます.
CTRL + M	↲	キーを押したのと同じ結果が得られます.
CTRL + L	CLS	キーを押したのと同じ結果が得られます.
CTRL + C		入力の終結を意味します.
CTRL + X		入力の終結を意味します.
CTRL + SHIFT + 0 (1)		オートリピート機能を停止(再開)します.

シフトキー（Shift key）

シフトキーは，他のキーと同時に押下することにより，次のような意味を持ちます．

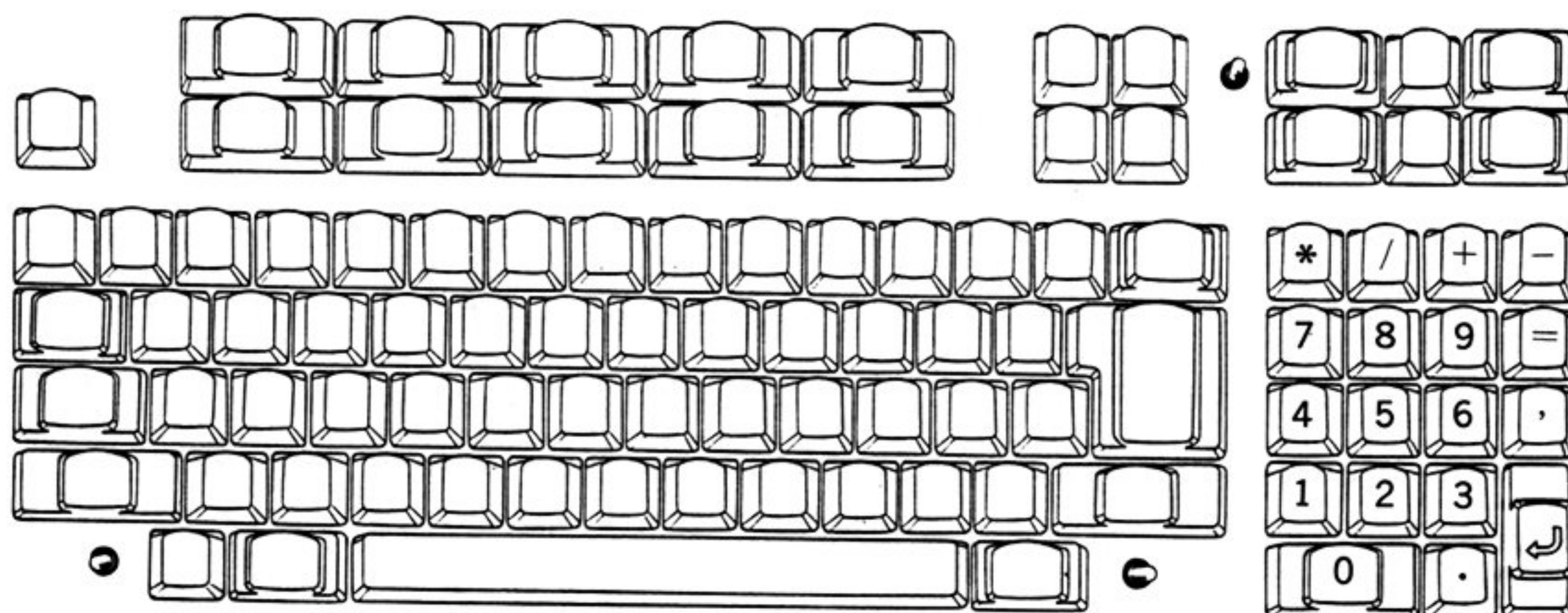
SHIFT ＋ ←	カーソルが左方向の英数字列の先頭へ移動します．
SHIFT ＋ →	カーソルが右方向の英数字列の先頭へ移動します．
SHIFT ＋ ↑	カーソルが前の行の先頭へ移動します．
SHIFT ＋ ↓	カーソルが後の行の先頭へ移動します．

その他，英文字モード，カナモード時におけるシフトキーの使用は，p. 5-3～5-5を参照して下さい．

5.3 エディットキー

キー	説明
INS インサート キー (insert key)	このキーが入力されると，挿入モードに設定され，INS キーの左側の状態表示ランプが点灯します．挿入モード時は，文字キーを押すと，カーソル位置にそのキーの文字が入り，カーソル以降の文字は1文字右へシフトし，その行がいっぱいになるまで行なうことができます． 挿入モード解除はもう一度 INS キーを押しますと，状態表示ランプが消え挿入モードが解除されます．
DEL デリート キー (delete key)	このキーが入力されると，カーソル位置の文字を削除し，カーソル以降の文字列全体を左へ1文字シフトします．
↑ ← → ↓ カーソル移動キー	これらのキーが入力されると，カーソルを行または，文字単位で指定された方向へ移動させます．いずれのキーも同一行または，同一桁でラップ（WRAP）します． ラップ……カーソルが最上位にある状態で ↑ キーを押すと，同一桁で最下位にカーソルが移動する．

EL	イレーズライン キー (erase line key)	このキーが入力されると，カーソルが点滅していたところより文字列が消去されます．
DUP	デュープ キー (dup key)	このキーが入力されると，カーソルは次のタブ位置へ移動します．このとき，カーソルの通過した部分は，1段上の文字が複写されます．ただし，カーソル位置が画面の最上段の場合は何もしません．
CLS	シーエルエス キー (cls key)	このキーが入力されると，現在の画面を消去し，カーソルを画面の先頭に移動します．
HOME	ホーム キー (home key)	このキーが入力されると，カーソルが画面の先頭に移動します．
⇦	バックスペース キー (back space key)	このキーが入力しますと，カーソルの左側の文字，1字を抹消します．

5.4 プログラマブル ファンクション キー (Programable Function Key)

これらのキーが入力されると，定義されている文字がキーインされたのと同じことになります．通常は次のように定義されていますが，ユーザはF-BASICのKEY文により再定義することができます．␣はスペースを意味します．

PF キー	ROM モード	DISK モード
PF1	AUTO␣	AUTO
PF2	LIST ↵	LIST ↵
PF3	RUN ↵	RUN ↵
PF4	CONT ↵	CONT ↵
PF5	LLIST ↵	LLIST ↵
PF6	LOAD ↵	LOAD"
PF7	SAVE"	SAVE"
PF8	?DATE$, TIME$ ↵	FILES
PF9	SCREEN 7, 7 ↵	SCREEN 7, 7 ↵
PF10	HARDC ↵	HARDC ↵

5.5 テンキー（Ten key）

キーボードの右下のキーの部分を総称してテンキーと呼びます．使用頻度の高い数字，演算記号を使い易いようにまとめたキーです．

0 ～ 9	ナンバー キー (number key)	数字が出力されます．
*	アスタリスク キー	積（かける）を意味する演算記号です．
/	スラッシュ キー	商（わる）を意味する演算記号です．
+	プラス キー	和（たす）を意味する演算記号です．
−	マイナス キー	差（ひく）を意味する演算記号です．
=	イコール キー	イコールを出力します．
,	コンマ キー	コンマを出力します．
.	ピリオド キー	ピリオドを出力します．
↲	リターン キー	このキーが入力されると，入力が終結されます．

5.6 カナキーの位置

① ア行

ア ウ エ オ
イ

② カ行

カ
キ ク ケ
コ

③ サ行

ス セ
シ
サ ソ

④　タ行

タ
テ
チ
ト
ツ

⑤　ナ行

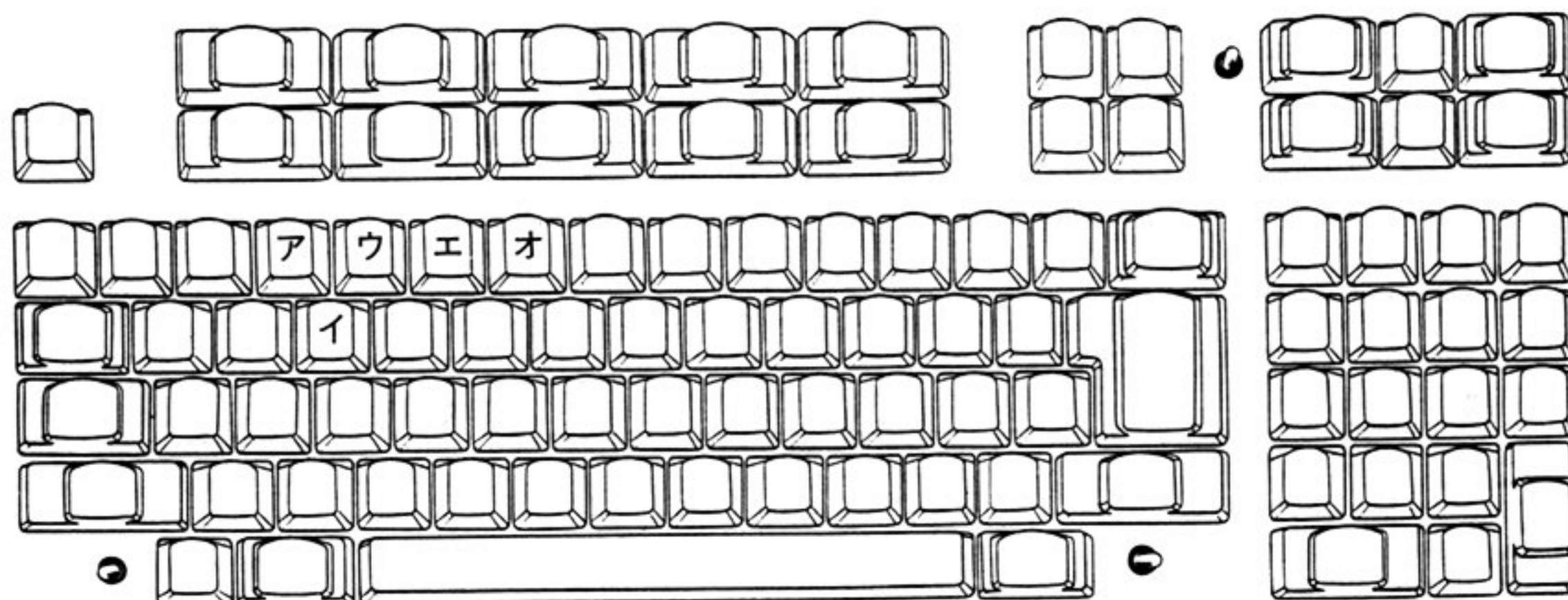
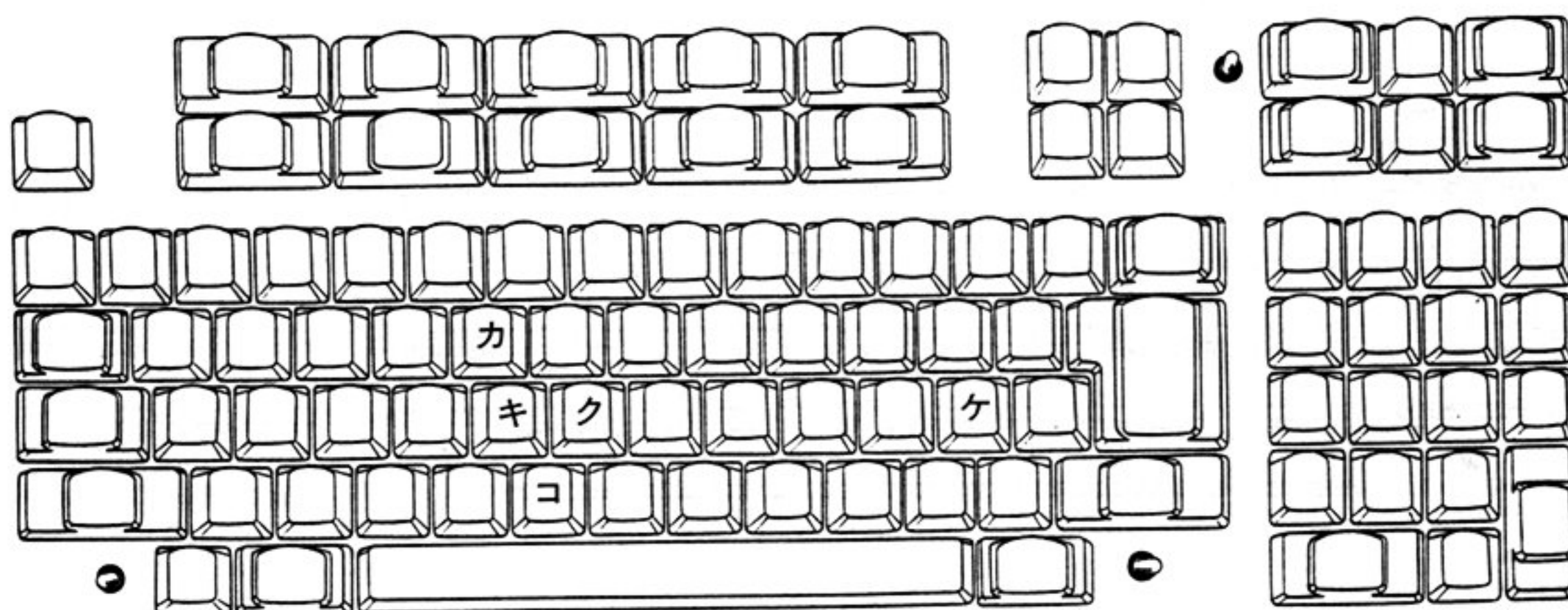
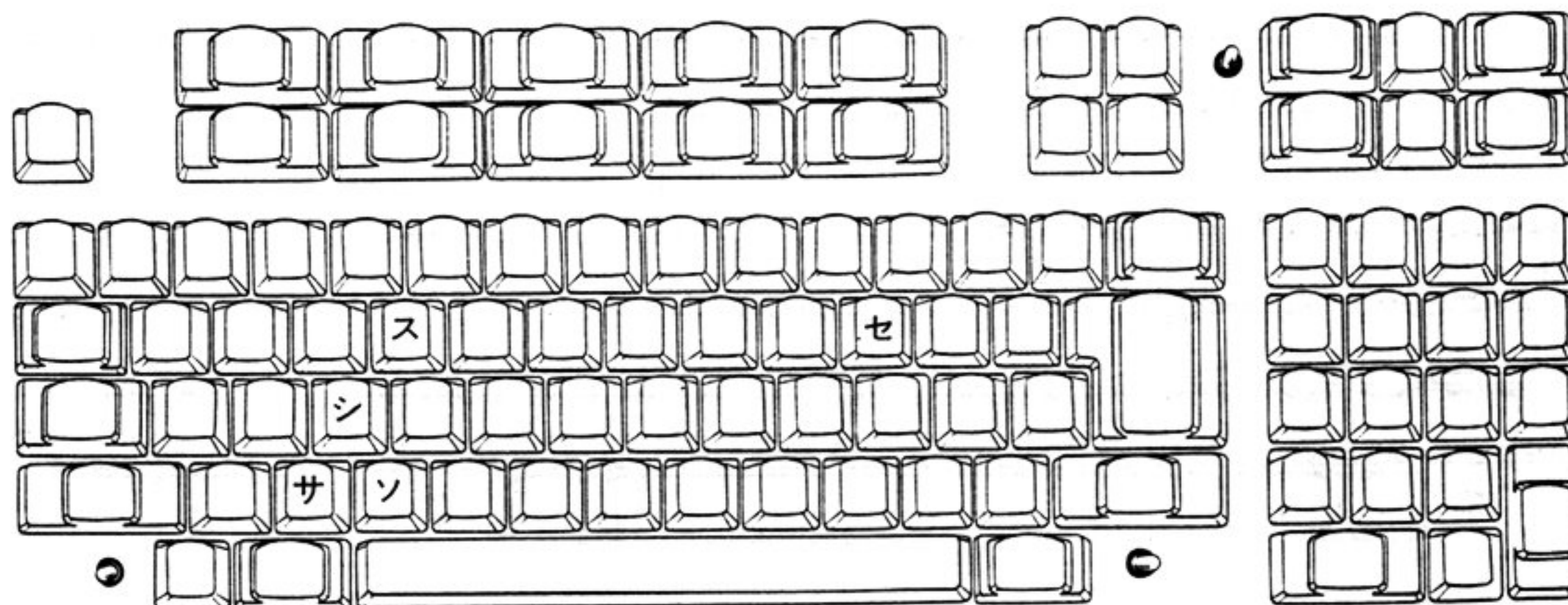
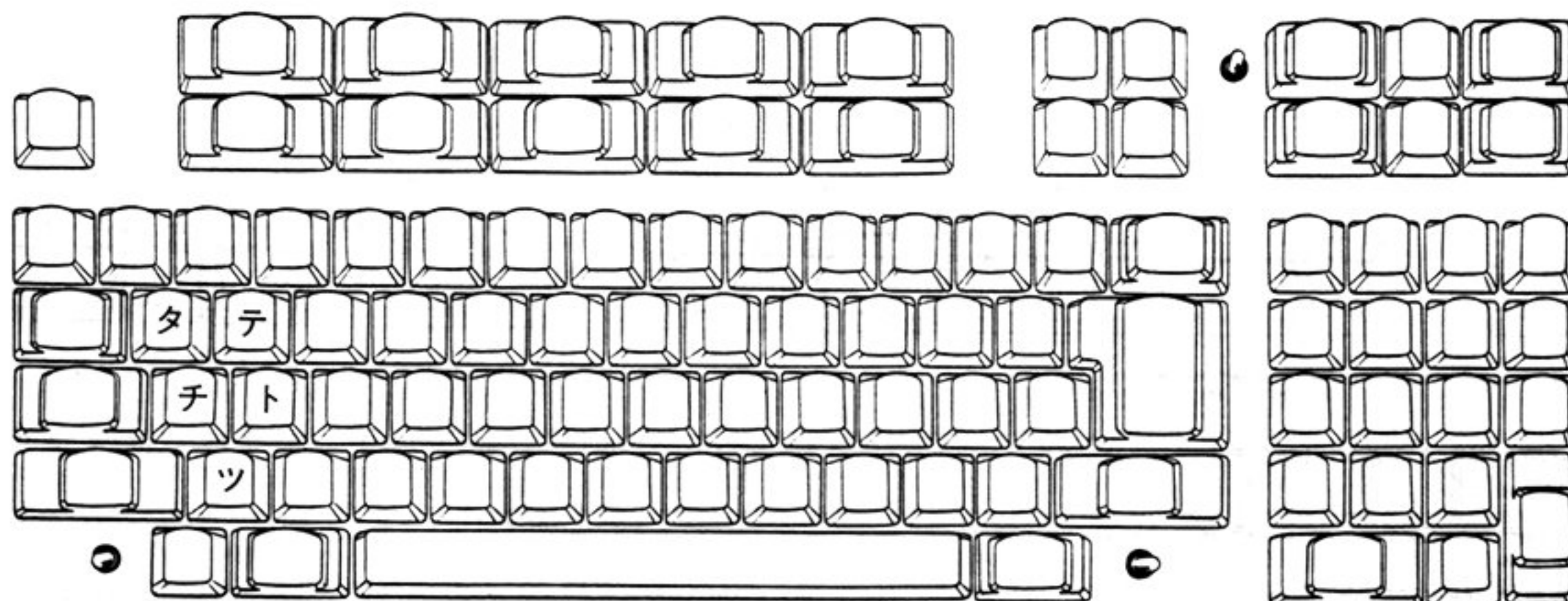
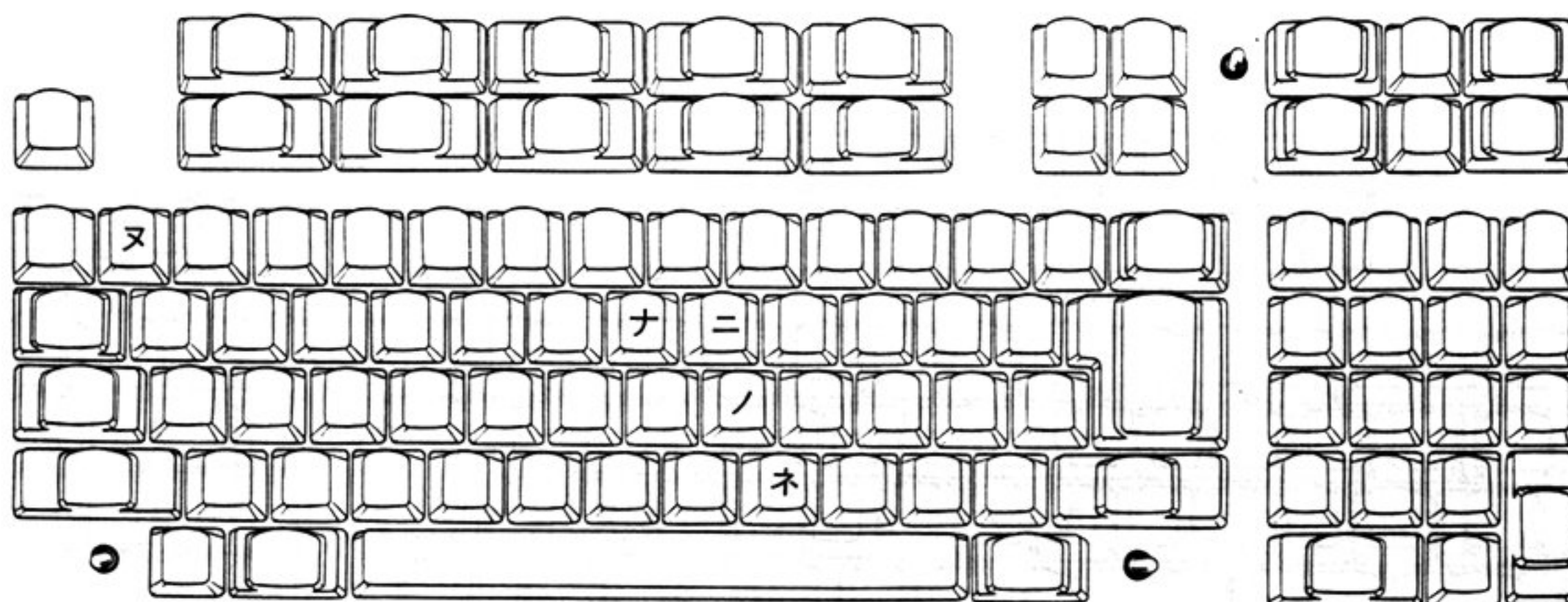
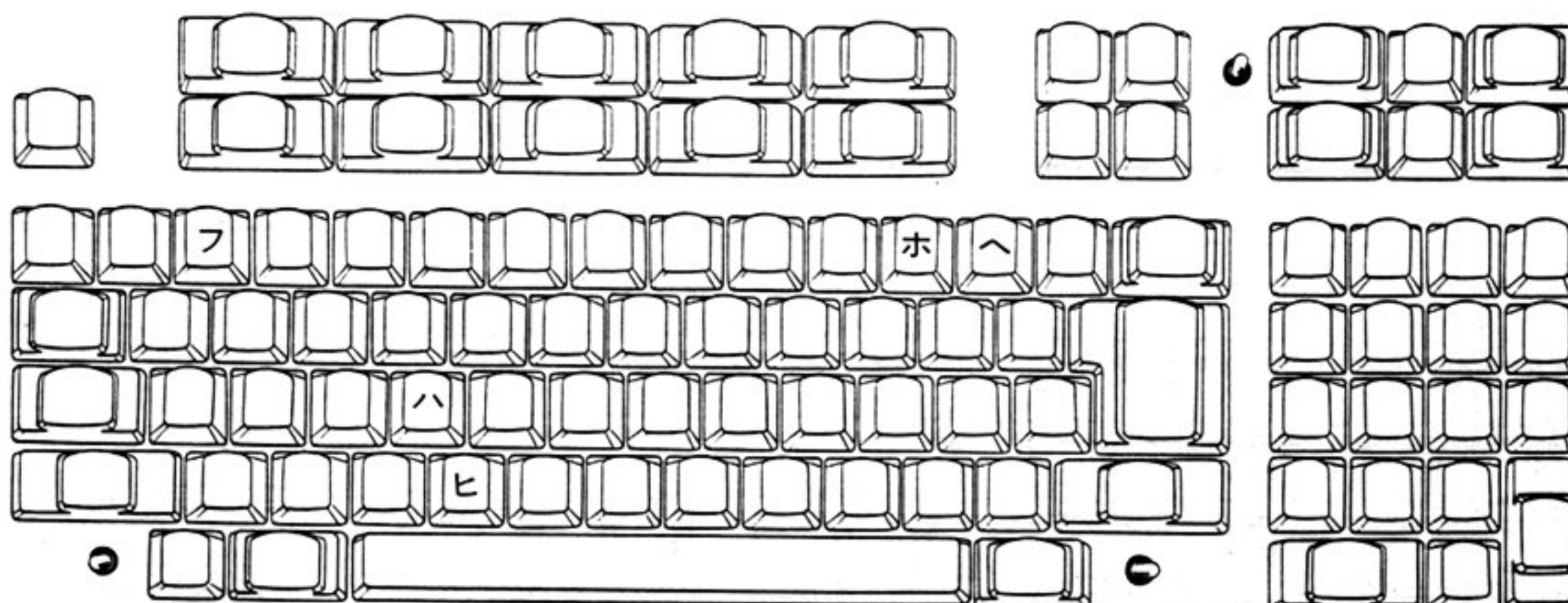
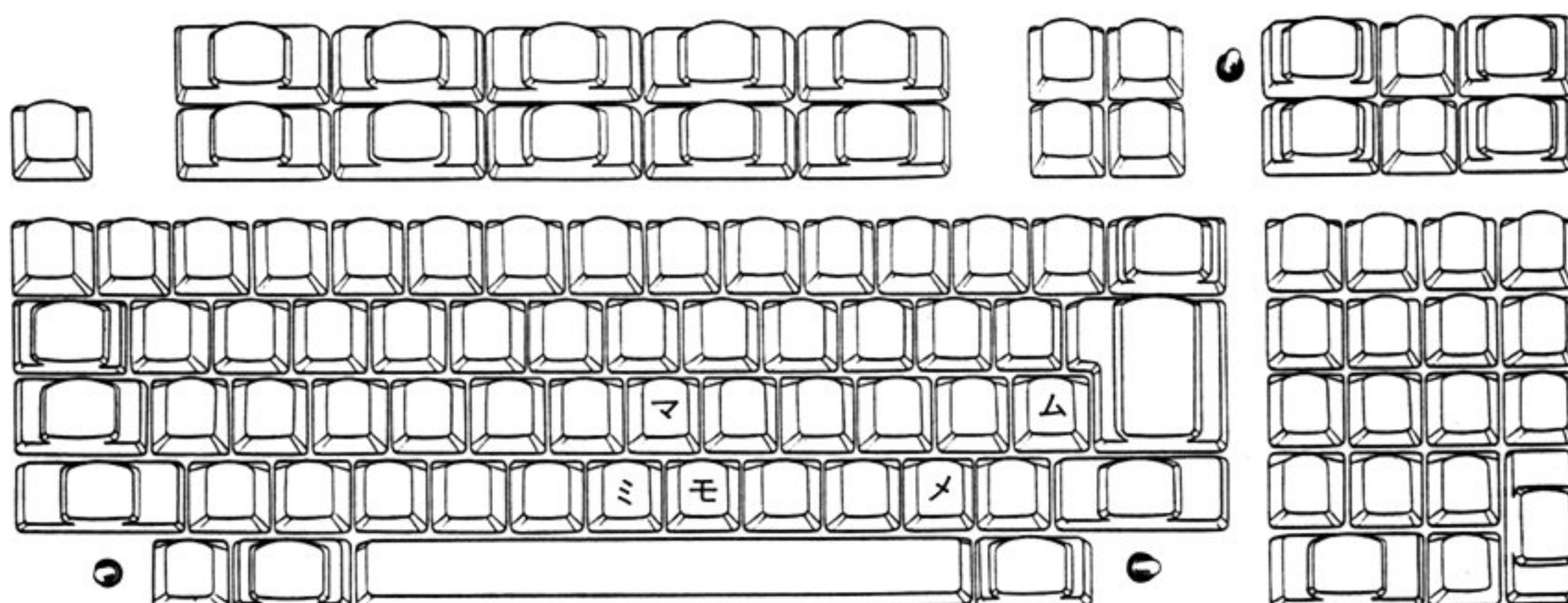
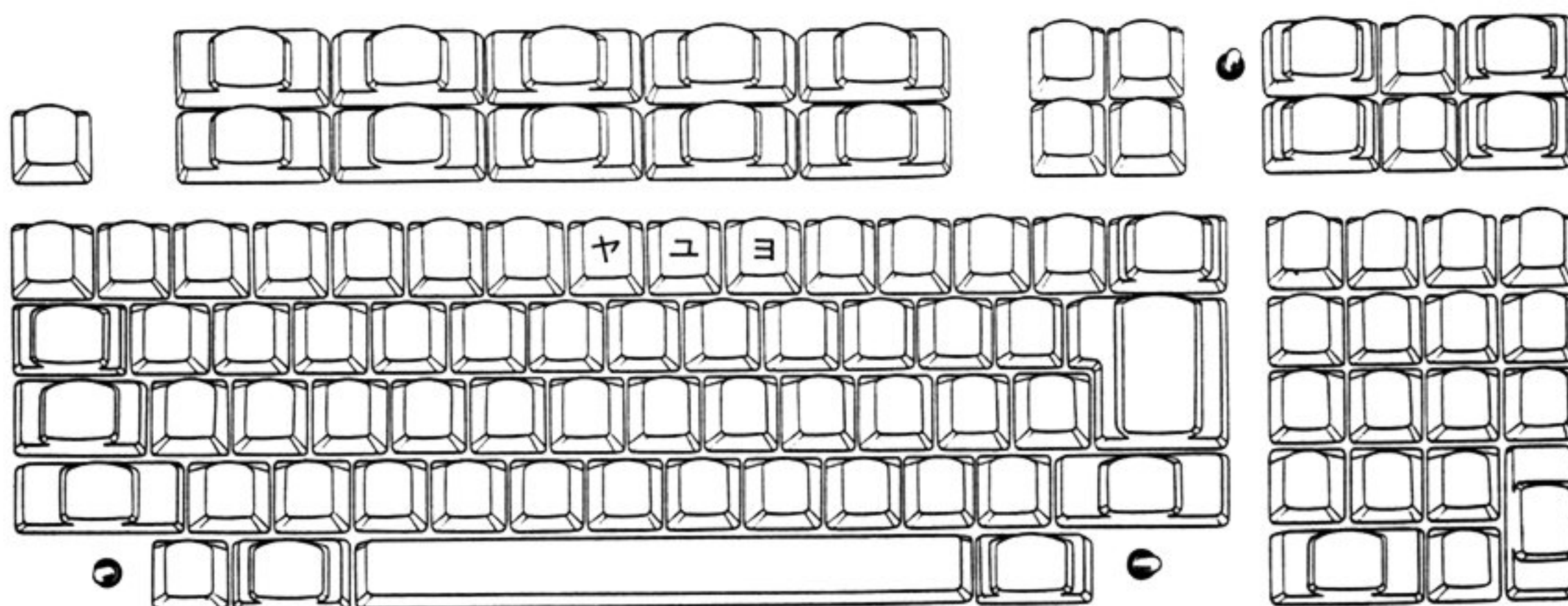
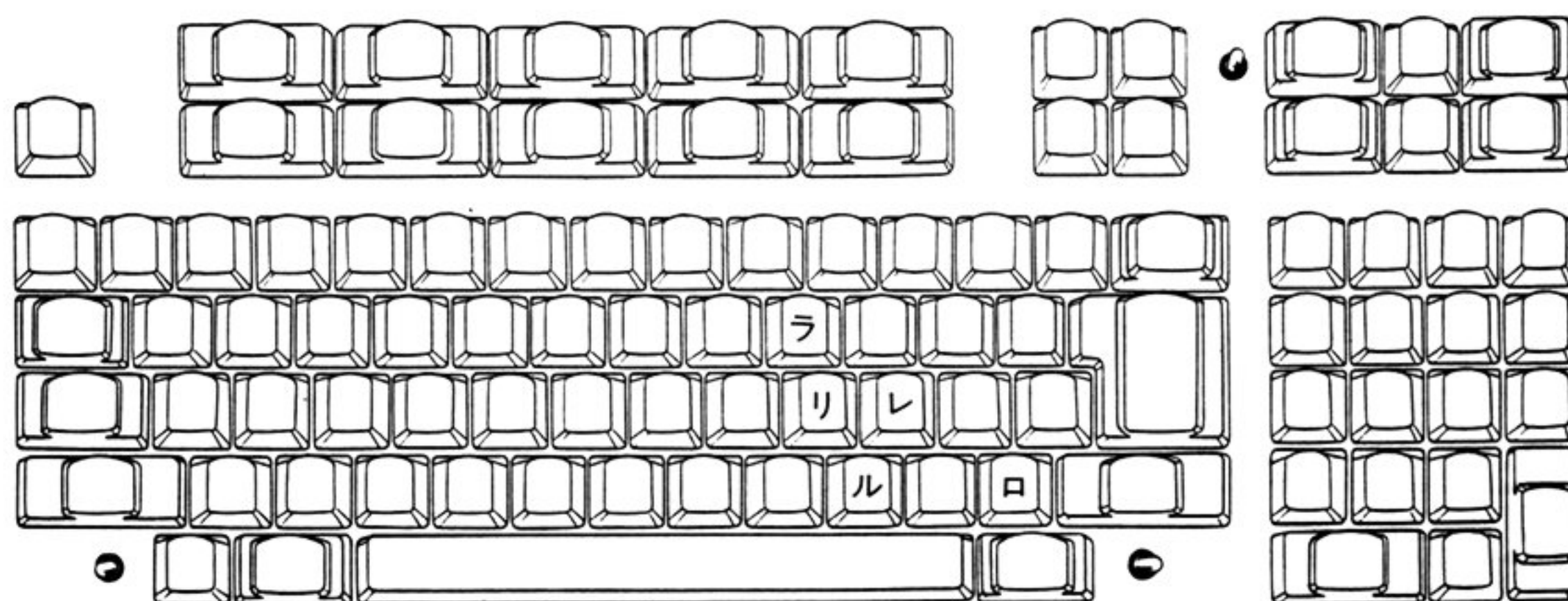
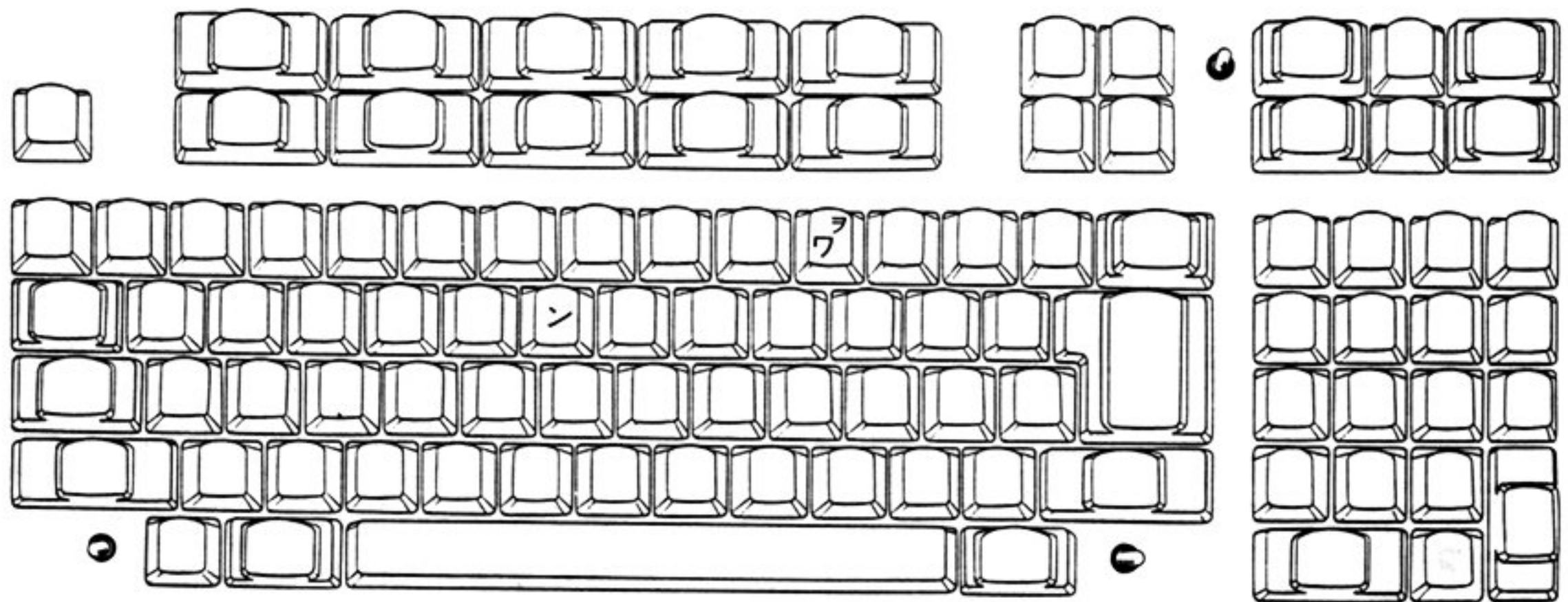
ヌ
ナ
ニ
ノ
ネ

⑥　ハ行

フ
ホ
ヘ
ハ
ヒ

⑦ マ行

⑧ ヤ行

⑨ ラ行

⑩　ワ行，ン，ヲ

第6章　プログラムの入力と修正方法

6.1　プログラムの入力方法

「プログラムの入力」という言葉は，カセットテープやフロッピィディスクからプログラムをロードするときにも用いますが，本項では，自分で作成したプログラムを，キーボード上の英数字等が彫刻された各種キーを押す（以下，キー入力と略します.）ことによって，メモリ内に記憶する方法について述べます.

6.1.1　AUTO（オート）コマンドによる方法（ROMモード）

[PF1]　　[PF1] [1] [0] [↵]

F-BASICのコマンドにAUTOコマンドがありますが，これは，プログラムを入力する際に自動的に行番号を発生させる働きをします.

一般的にコマンドを入力するには，各アルファベット（この場合は，AUTO）をキー入力しますが，FM-7では，このコマンドを[PF1]キーに，定義してありますので，[PF1]キーを押すことで，

[A] [U] [T] [O] [␣]

と，キー入力したと同じ状態になります.

[PF1]の最後の〝␣″（スペース）は，1文字あけることを意味し，キーボード手前の8キー分の長さを持つキーを1回押しますが，〝AUTO␣″まで，PF1キーに定義されておりますので，この場合は，キー入力はしません.

[0]のキー入力は，「行番号の始まりを10にする」という意味です．[1]，[0]は，キーボード上テンキーのもの，フルキーの中の[1!ヌ]，[0ヲワ]をキー入力，いずれの場合も同じで入力結果が得られます.

[↵]（リターンキー）がキー入力されると，FM-7は，AUTOコマンドの実行を行ないます．リターンキーも数字キーと同様に，テンキー，フルキーのいずれのキー入力も同じ働きをします.

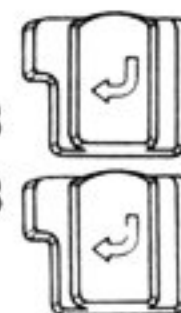
CAP

CAP （キャピタルキー）を押して，左側の状態表示ランプが点灯している状態で英字キーを押すと，英大文字がキー入力されます．

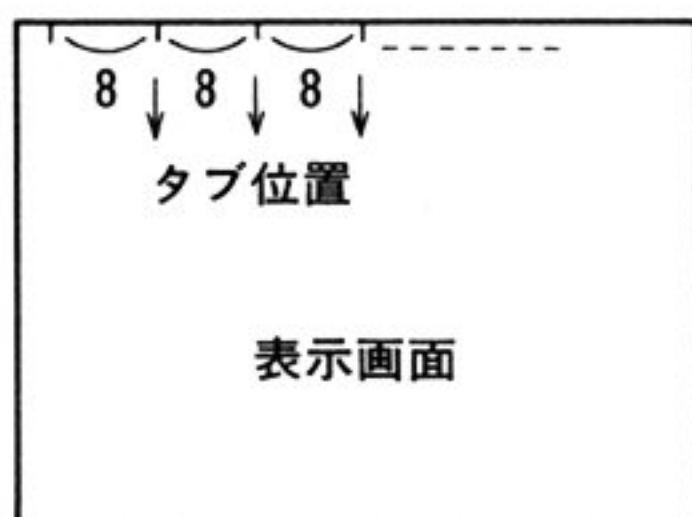
CAP キーが押されていない状態（ランプが消えている）で，SHIFT キーを押しながら英字キーを押した場合も同様に，英大文字が入力されます．

英字キーのみのキー入力では，英小文字の入力となります．

プログラムの入力

```
10 LINE (0, 0)-(639, 199), PSET, 6, B ↵
20 LINE (0, 0)-(629, 189), PSET, 6, B ↵
30
```

ここで，DUP （デューフキー）を2回押して下さい．

```
30 LINE (0, 0)-(6
```

8 ↓ 8 ↓ 8 ↓
タブ位置
表示画面

このように表示されます．

DUP キーは，1行上の文字をコピーする機能を持っています．

コピーする文字は，入力位置を示す■（カーソル）が点滅する位置より，画面表示位置の左端から右端までを8文字分ずつ区切ったタブ（TAB）位置までの文字を行ないます．

現在，カーソル位置が〝6〞の右横にあるはずです．ここから数字，カンマ，右カッコまで，下に記したようにキー入力して下さい．

```
30 LINE (0, 0)-(619, 179)
```

あとは，上の行と同様の文字であるので，DUP キーを押します．

1回キー入力すると〝,〞を表示するのみです．（ここもタブ位置です．）もう1度 DUP キーをキー入力して下さい．

〝B〞までコピーされたことを確かめて，↵ キーを入力して下さい．

以下，同様に行番号90まで，カッコ内の数値を変えながら，プログラムを入力して下さい．

行番号50では，DUP を2回キー入力すると，〝6〞まで入力されてしまい

（ ）の入力方法：SHIFT キーを押しながら 8（ユ ，9）ヨ を押す．

，の入力方法：フルキー部の <，ネ または，テンキー部の ， を押す．

－ の入力方法：フルキー部右上部の ニ－ホ または，テンキーの － を押す．

ます.

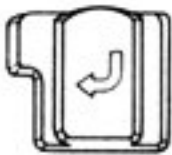
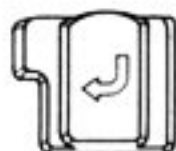
このような場合，あるいは，まちがった文字をキー入力してしまった場合には，[⇦]（バックスペース：フルキー部右上）キーを押して下さい.

[←]キーは，カーソル位置の1文字前の文字を削除します．削除した後のカーソル位置は，削除前より1文字左へもどります.

```
40  LINE (0, 0)-(609, 166), PSET, 6, B [↵]
50  LINE (0, 0)-(599, 159), PSET, 6, B [↵]
60  LINE (0, 0)-(589, 149), PSET, 6, B [↵]
70  LINE (0, 0)-(579, 139), PSET, 6, B [↵]
80  LINE (0, 0)-(569, 129), PSET, 6, B [↵]
90  LINE (0, 0)-(559, 119), PSET, 6, B [↵]
```

現在，カーソル位置の行番号は，100のはずです.

今までキー入力してきた方法を確かめて，以下100〜140のプログラムをキー入力して下さい.

{なお，「"」（ダブルクォーテーション）は，[SHIFT] キーを押しながら，[2 " フ]を，または，キーボード右下の[" ロ] キーのみを押します.}

```
100  SYMBOL (5, 2), "FUJITSU", 6, 4, 3 [↵]
110  SYMBOL (75, 45), "PERSONAL", 6, 4, 5 [↵]
120  SYMBOL (150, 90), "COMPUTER", 6, 4, 7 [↵]
130  GOTO 130 [↵]
140  END [↵]
```

AUTOコマンドの終了

現在，行番号が150ですがAUTOコマンドの終了は，行番号の右横にカーソル位置があるときに，[↵]キーを押すことにより行ないます．画面には，Readyと表示されます.

このReadyと表示している状態をコマンドレベルと呼び，各種コマンドがキー入力して実行可能なことを示します.

6.1.2 行番号入力による方法

プログラム入力の際，AUTOコマンドにより自動的に行番号を発生させないで，そのつど，

　行番号＋文

とキー入力をする方法です．

プログラムの修正，追加等のデバック（debug）作業の際には，頻繁に使用します．

さきほどまで入力したプログラムに1文だけ追加します．

（コマンドレベルにあることを確認します．）

　5 CLS [↵]

以上でプログラムのキー入力は終了しました．

さっそく実行してみましょう．

[PF3]

[PF3] キーをキー入力して下さい．

FUJITSU
PERSONAL
COMPUTER

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

左図のような絵が出なかった場合には，キー入力を間違っているのです．

その時には，以下のようにキー入力してから，6-1ページにもどって，プログラムを入れ直して下さい．

1. [BREAK]　（キーボード左上部）
2. [N] [E] [W] [↵]

これは，NEW（ニュー）コマンドによって，メモリ上に記憶されているプログラムをすべて消すという作業を行ないました．

6.1.3 プログラムの実行

前ページの実行で [PF3] キー入力をしましたが，これは RUN（ラン）コマンドによって，実行したわけです．

[PF3] キーの入力は，AUTO コマンドと同様に，

[R][U][N] [↵]

が定義づけられているので，「RUN」コマンドを実行したことになります．

RUNコマンドは，プログラムの途中から実行させることも可能です．

(例) [R][U][N] [1][0][0] [↵]

いずれの場合もコマンドレベルにある場合のみ実行可能です．

6.1.4 プログラムリストの表示

メモリ内に記憶されているプログラムを画面上に表示する場合には LIST（リスト）コマンドを用います．

[BREAK][CLS][PF2]

FUJITSU
PERSONAL
COMPUTER

現在，6.1.1 および 6.1.2 によって入力されたプログラムは，実行されたまま（プログラム上で，終了しないようなものになっている．）の状態であるので，以下の順序でリスト表示をします．

(1) [BREAK] キー（実行の中断）

(2) [CLS] キー（全画面の消去）

(3) [PF2]（LIST コマンド）

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

左図の通りになっているはずです．

LIST コマンドは，以下のように1文，1部の文のリスト表示も可能です．

(例) LIST 20 [↵]

LIST 20-50 [↵]

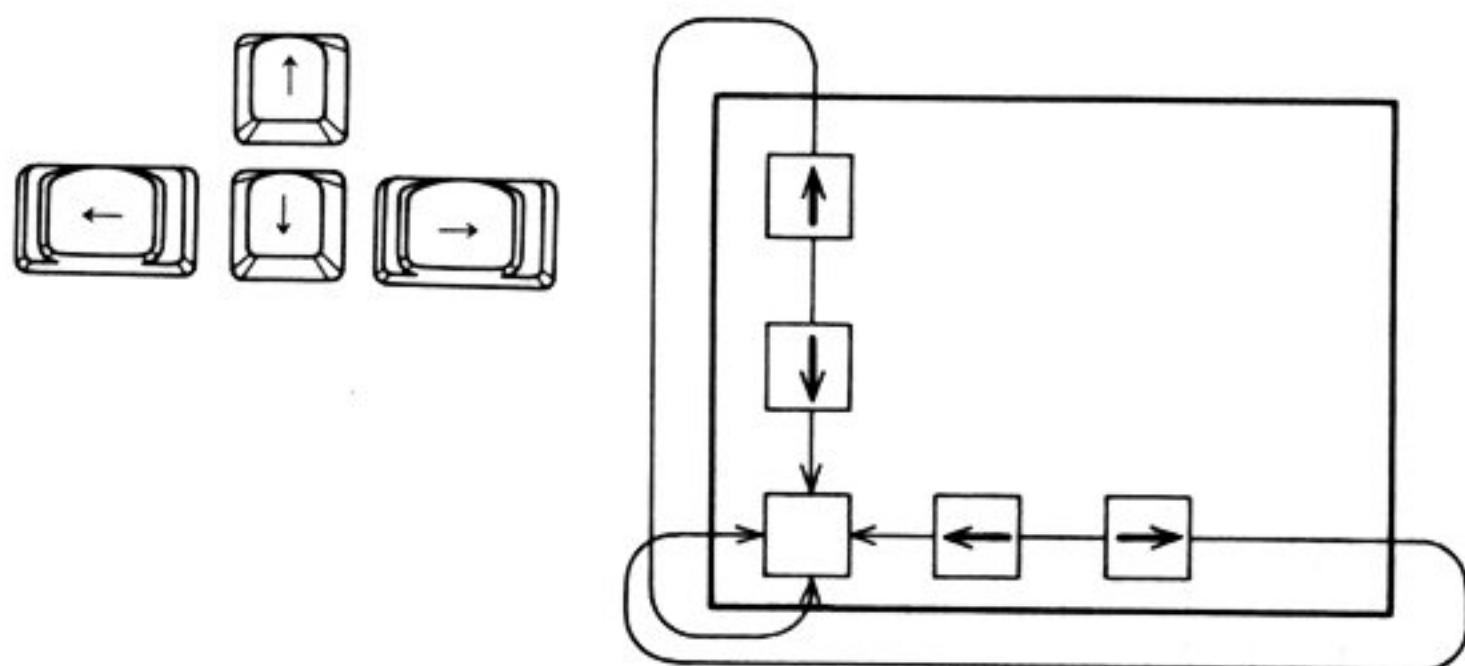
6.2 プログラムの修正方法

FM-7でのプログラムの修正は，メモリ上に記憶されたプログラムを，画面上にリスト出力して行ないます．

このような，画面上でプログラム等の修正，追加等の編集を行なう機能をスクリーン・エディタ(screen editor)と呼びます．

スクリーン・エディタは，画面上の文字を，カーソル移動キーにより指定して，編集を行ないます．

スクリーン・エディタの機能は，[↵]キーが押されるまで有効となります．

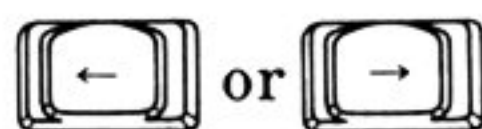
6.2.1 カーソル移動キー

カーソルは，スクリーン・エディタにおいて，修正する位置を示す重要な表示です．

FM-7のキーボードでは，上下，左右の4方向のカーソル移動キーを有しています．

この4つのキーによるカーソルの動きは，キー表面に彫刻された「矢印」と画面上の動きと対応します．

同じカーソル移動キーを押し続けると，キー読取りのリピート(repeat)機能によって，カーソルは移動しつづけます．

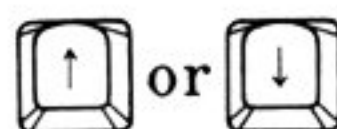
上下左右のいずれかの端に移動すると，カーソルは，左図の如く上下左右に回転するよう移動します．

カーソル移動キーは，[SHIFT]キーと併用([SHIFT]キーを押しながら，カーソルキーを押す．)すると単独で使用するときとは異なった動きをします．

SHIFT +
← or →

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

(1) 横方向移動キーとの併用

カーソルは，現在の位置より，左もしくは右方向の英文字列の先頭へ移動します．

SHIFT +
↑ or ↓

HOME

(2) 縦方向移動キーとの併用

カーソルは，現在の位置より，前あるいは後の行の先頭に移動します．

また，HOME（ホーム）キーは，カーソルを画面左上のホーム・ポジション(home position)に移動させます．

6.2.2 文字の訂正

[↑] [→] [←]

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

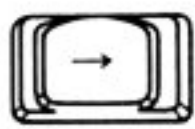
(1) [↑] キーを使用して，カーソル位置を，行番号100の行に移動する．

次に，[→] or [←] キーによって〝F〞の文字にカーソルを移動する．

ここで，「FUJITSU」の文字のかわりに「ABCDEFG」と順番にキー入力して下さい．

最後に，[↵] キーを押す．
（文字の訂正の終了）

[PF3]

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"ABCDEFG",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

ABCDEFG
PERSONAL
COMPUTER

入力の間違いがないときには，実行すると左図のように表示されます．

（元の字数よりも多い，または少ない字数にする場合は，次ページの挿入，削除の項を参照して下さい．）

[BREAK] [CLS] [PF2]

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU",3,2,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

(2) リストを表示する

上の逆に，行番号100の文中の「ABCDEFG」を「FUJITSU」にもどして下さい．

次に，[→] キーを使って〝6〞の位置にカーソルを移動させます．

[3] [,] [2]

（フルキー，テンキーいずれも可）

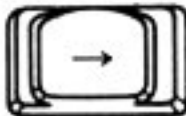
とキー入力します．

最後に，[↵] キーを押す．
（文字の訂正を終了）

[PF3]

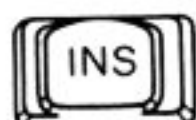
FUJITSU
PERSONAL
COMPUTER

実行してみましょう．（表示文字の大きさが変化）

6.2.3 文字の挿入

[BREAK] [CLS] [PF2]

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU",3,2,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

[INS]

[BREAK] [CLS] [PF2]

と順にキー入力してプログラムリストの表示をします．[↑] および [→] キーを用いて，行番号100の「"FUJITSU"」の後の「"」の位置にカーソルを移動して下さい．

ここで，キーボード右上部にある [INS] （インサート）キーを押して下さい。このキーにも状態を示すランプが左横にあり点灯します．

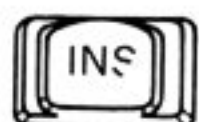
つぎに，以下のようにキー入力して下さい．

[␣] [M] [I] [C] [R] [O] [␣] [7] をキー入力したときに，「ピー」とブザーが鳴ります．
（このブザーについては，後で説明します．）

文字が追加され，全体が右にずれます．

このように，文字の挿入は，挿入可能状態を示すランプが点灯しているとき，挿入位置の次の文字にカーソルを移動させ，追加文字をキー入力することにより行ないます．

（注意）　ある箇所の挿入を終了して，挿入状態としたまま，別の文の文字訂正を行なおうとカーソルを移動してキー入力すると，挿入になってしまうので注意して下さい．（ランプの消去を確認）

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU MICRO 7",3,2,3
110 SYMBOL (175,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

つづいて，別の行の挿入を行ないます．

[↓]および[←]等のカーソル移動キーにより，行番号110「(75, 45)」の7の文字に，カーソルを移動します．(左のリストは，挿入後です．)

● [1]をキー入力して下さい．

●〝1″が挿入され，全体が1文字右にずれます．

[INS]

● [INS]キーを押してランプが消えることを確認します．(挿入終了)

[↵]

● [↵]を押して下さい．

(スクリーン・エディト終了)

●コマンドレベルとなります．

●実行してみましょう．

[PF3]

[PF3]をキー入力します．

●左図のように表示されます．

FUJITSU MICRO 7
PERSONAL
COMPUTER

挿入モードによって，不要な文字までキー入力してしまったときは，6.2.4 文字の削除の項を読んで修正して下さい．

文字を挿入の際のブザー音

さきほどの例のように，スクリーン・エディタにて文字を挿入する際，右端まで文字が入力されると，ブザー音が鳴り，キー入力を受け付けなくなります．

こうした場合には，以下のように対処して下さい．

まず，[↵]キーを押し，コマンドレベルにします．

対処方法

(1) LIST コマンドによりプログラムリストを再度表示する．

右端まで入力された文の下が1行分あけられます．

カーソルを移動し，挿入の続きを行なう．

(2) LIST コマンドにより挿入対象の文のみ表示する．

LIST 文番号 [↵]

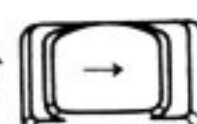
指定された文番号の内容が表示される．カーソルを移動し，挿入を続行する．

(3) EDIT コマンドにより行番号を指定して表示，挿入を行なう．

6.2.4 文字の削除

BREAK CLS PF2

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FUJITSU MICRO 7",3,2,3
110 SYMBOL (175,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

BREAK CLS PF2 と順にキー入力してプログラムリストの表示をします.

↑ および → キーを用いて，行番号100の「"FUJI…"」の「U」の位置にカーソルを移動して下さい.

ここで，キーボード右上端にある DEL（デリート）キーを7回押して下さい.

DEL

「UJITSU␣」の文字が消え，以降の文字が左につめられます.

カーソルを1つ進めて，「I」をカーソル位置として，DEL キーを4回押します.

そこで −(ホ) をキー入力します.

カーソルを3つ右に進めて，6 ， 4 の順にキー入力します.

つづいて，行番号110の"1"を削除して下さい.

DEL キーは，このように，カーソルで指定した位置の文字等削除して，以降の文字を左へつめる動作をします.

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

- ↵ キーを押して下さい.（スクリーン・エデイット終了）
- コマンドレベルとなります.
- 実行します.

PF3

PF3 をキー入力します.

- 左図のように表示されます.

FM-7
PERSONAL
COMPUTER

6.2.5 文の挿入

BREAK CLS PF2

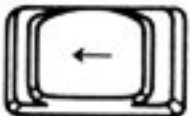
```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

FM-7
PERSONAL
COMPUTER
FUJITSU

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
15 SYMBOL (450,191),"FUJITSU",3,1,2
```

現在，左図のリスト表示のプログラムがメモリ内に記憶されています．ここで，プログラムの途中に，別の文を入れる方法について説明します．

左図のように右下の位置に「FUJITSU」という文字を表示する文を追加する．

SYMBOL␣(450, 191), "FUJITSU", 3, 1, 2 ↵

を行番号15として入力することにより表示されます．

（行番号は，5～130の使用していない番号であればかまいません．）

BASICによるプログラムの行番号は，こうした文の追加等のために，一般的に10ずつあけた行番号をとっていきます．

(1) 行番号からキー入力する

コマンドレベルにあることを確認して，左図のように，行番号15からキー入力します．

（入力を間違えたときは，← キーを使用して訂正します．）

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
15 SYMBOL (450,191),"FUJITSU",3,1,2
20 LINE (0,0)-(629,189),PSET,6,B
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

(2) プログラム中の文を使用

● カーソルを行番号120の [1] に移動させます.

ここで, [1] [5] とキー入力して, カーソルが "0" に移動したところで, [DEL] キーを1回押して, 内容が行番号120と同じものを, 行番号15として記憶する.

● [↵] キーを1回押す.

● [L] [I] [S] [T] [␣] [1] [5] [↵]

で, 行番号15のリストを表示する.

カーソルを移動して, 各種エディタキーを用いて, 希望の文とする.

● 実行します.

[PF3]

[PF3] をキー入力します.

FM-7
PERSONAL
COMPUTER
FUJITSU

● 希望の絵が表示されているか確認して下さい.

● 希望通りに表示されない場合は,

(1) リスト表示

(2) 修正

(3) 実行

をくり返して下さい.

6.2.6 文の削除

BREAK CLS PF2

```
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

EL

```
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
15
20
30 LINE (0,0)-(619,179),PSET,6,B
40 LINE (0,0)-(609,169),PSET,6,B
50 LINE (0,0)-(599,159),PSET,6,B
60 LINE (0,0)-(589,149),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
DELETE 30-50
```

プログラムを作成（修正）している際には，入力した文が不要となることがよくあります．

この場合の文を削除するいくつかの方法について説明します．

「15～60までの文を削除する．」

(1) EL（イレーズライン）キーによる方法．

●目的の行番号の文の先頭文字（行番号15では「S」の位置）にカーソルを移動する．

● EL キーを押す．

●以下の文が削除されます．

●同様に，20の文も，「L」の位置にカーソルを移動して，EL キーを押す．

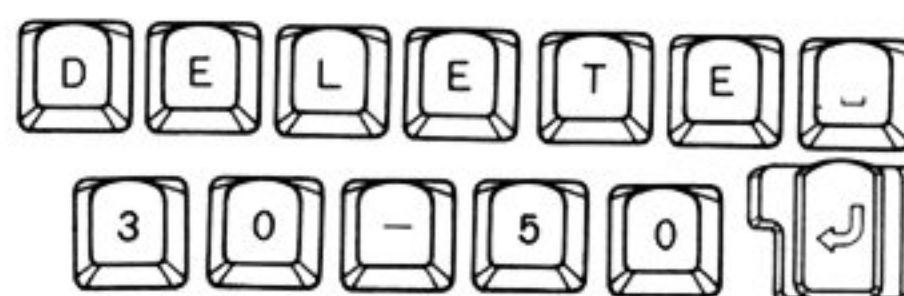
● ↵ キーを押す．（スクリーン・エディト終了）

EL キーは，このように，不要な文，あるいは，文の途中からそれ以降が不要といったときに用います．

(2) DELETE（デリート）コマンドによる方法

●コマンドレベルにあることを確認します．

●以下のようにキー入力します．

D E L E T E ␣
3 0 - 5 0 ↵

（30～50までの文を削除）

```
5 CLS
10 LINE (0,0)-(639,199),PSET,6,B
70 LINE (0,0)-(579,139),PSET,6,B
80 LINE (0,0)-(569,129),PSET,6,B
90 LINE (0,0)-(559,119),PSET,6,B
100 SYMBOL (5,2),"FM-7",6,4,3
110 SYMBOL (75,45),"PERSONAL",6,4,5
120 SYMBOL (150,90),"COMPUTER",6,4,7
130 GOTO 130
140 END

Ready
```

```
FM-7
  PERSONAL
    COMPUTER
```

(3) 行番号をキー入力して削除する方法

●コマンドレベルにあることを確認します.

●目的とする行番号をキー入力します.

6 0 ↵

(60の文を削除)

ここで, PF2 キーでリストを表示すると左上図のようになります.

文の削除は, こうした方法によって行ないます.

6.3 その他

プログラムの作成，修正の際に用いるその他のキーの説明をします．

6.3.1 ESC （エスケープ）キー

●「LIST」コマンド等によって，長いプログラムを画面に表示して，プログラムの確認作業を行なうときには，希望の行番号が表示されたところで，ESC キーを押すと，画面のリスト表示は一時停止します．

●リスト出力の継続は，BREAK キーを除く，いずれかのキー入力によりなされます．

また，一時停止している状態で，もう一度 ESC キーを押すと，次の行を表示します．

●一時停止した状態で，修正箇所が存在したときには，BREAK キーを押し，コマンドレベルにします．そして，スクリーン・エディトにより修正を行ないます．

6.3.2 PF1 ～ PF10 （プログラマブルファンクションキー・または，ピー・エフキー）の設定

F-BASIC の文またはコマンドを PF キーに設定する事により，そのたび事にキーボードから文字を入力する事なく PF キーを1回押すことで入力ができるようになります．

F-BASIC V3.0 では電源投入時には次のように定義されていますが F-BASIC の KEY 文により，新たな内容を定義できますので，他の処理への使用が可能です．

●ROM モード時

PF 1	AUTO␣	PF 6	LOAD CR
PF 2	LIST CR	PF 7	SAVE "
PF 3	RUN CR	PF 8	?DATE$，TIME$ CR
PF 4	CONT CR	PF 9	SCREEN 7，7 CR
PF 5	LLIST CR	PF 10	HARDC CR

●DISKモード時

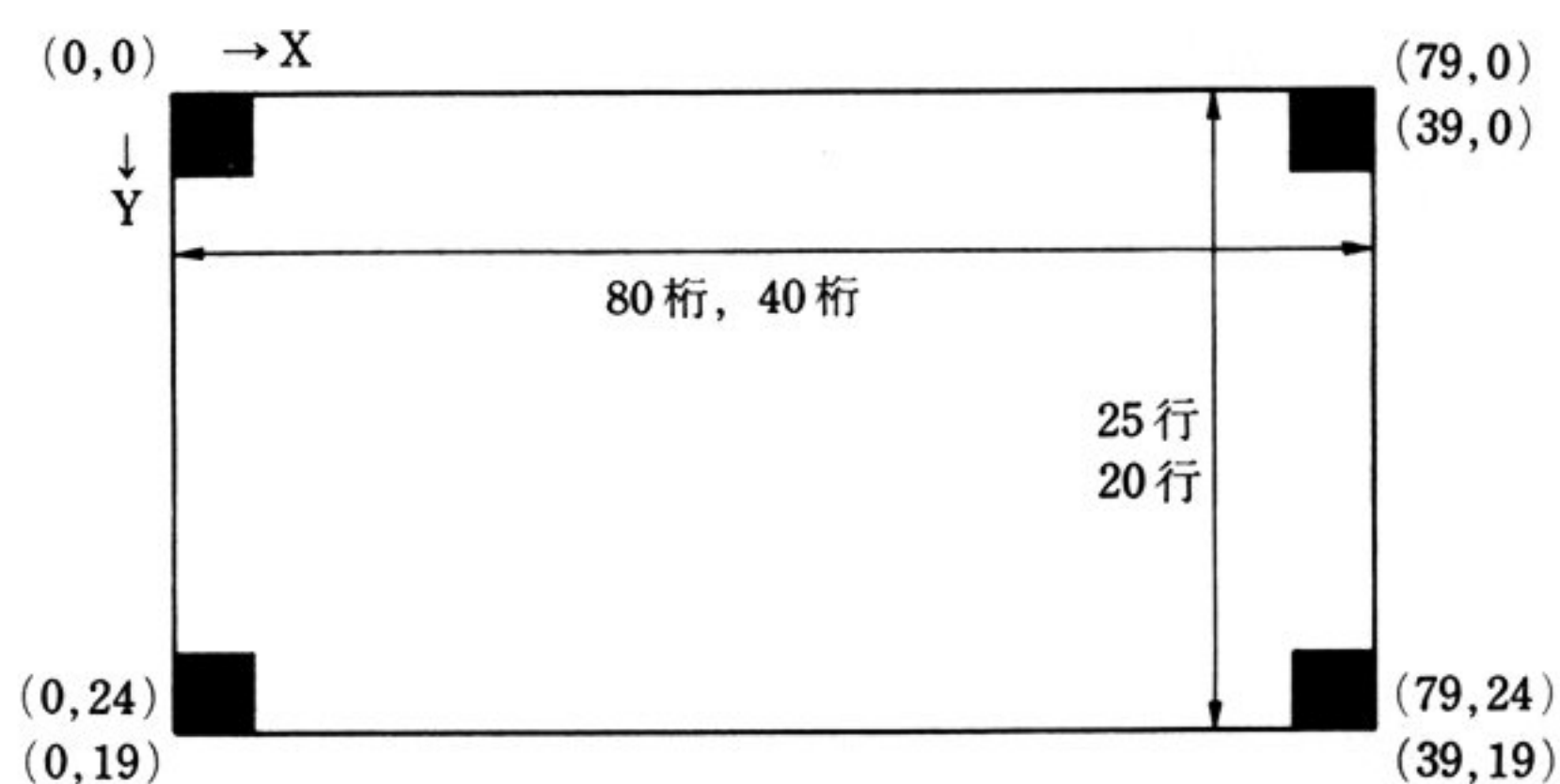
PF1	AUTO␣	PF6	LOAD"
PF2	LIST CR	PF7	SAVE"
PF3	RUN CR	PF8	FILES
PF4	CONT CR	PF9	SCREEN 7,7 CR
PF5	LLIST CR	PF10	HARDC CR

␣はスペースを表わし，CRはプログラマブルファンクションキーが押された時点で，すぐに文字列を実行することを意味します．

第７章　ディスプレイ表示

7.1　キャラクタ表示

7.1.1　FM-7のキャラクタ座標

(0,0)　→X
↓
Y
(79,0)
(39,0)
80桁，40桁
25行
20行
(0,24)
(0,19)
(79,24)
(39,19)

このように，キャラクタ表示は，横80桁または40桁，縦25行または20行の選択機能を持っており，それぞれを組み合せることが可能です．

7.1.2　表示文字数の特長

方向	桁数	特長
横	40	文字の大きさが大きい
	80	文字の大きさが小さい
縦	20	上下の文字間隔がはなれている
	25	上下の文字間隔が近づいている

・同じディスプレイ上に，40桁，80桁と表示できますが40桁の方が文字が大きく見やすくなります．
・縦の20行，25行は，文字の大きさは変わらず，文字の上下間隔があくか，つまるかの違いです．

7.1.3 各モードによるディスプレイの状態

種類 ＼ モード		40×20	40×25	80×20	80×25
単色	グリーン CRT	○	○	○	○
	家庭用白黒 TV*	○	△	×	×
カラー	カラー CRT	○	○	○	△
	高解像度 CRT	○	○	○	○
	家庭用カラー TV*	○	△	×	×

○推　奨，△推奨しないが使用不可能ではない，×非推奨

*家庭用カラーテレビアダプタが必要です．

7.1.4 表示文字数の変更（WIDTH 命令）

WIDTH　横方向の桁数，縦方向の行数

表示文字数の変更は WIDTH 文で行ないます．初期セット時は，WIDTH 40，20 にセットされています．必要に応じて行数，桁数を設定して下さい．

(例)　WIDTH　80，25

7.1.5 キャラクタ表示

キャラクタ表示に関するいくつかの BASIC の命令について解説します．

詳細は，F-BASIC 文法書を参照して下さい．

(1) 画面上に，ファンクションキー（PF キー）の内容を表示するときは CONSOLE 命令を使います．

CONSOLE　0，(現在の表示行数－2)，1

を行なうと，PF キーに対応して，PF キーの内容を画面上の下部 2 行に表示します．

(2) 画面に文字を書くときは PRINT 文を使います．

画面上に，たとえば FM-7 と書くには，

PRINT　"FM-7"

これを実行すると，次の行に FM-7 と表示されます．また，PRINT の代りに ? を使うことも可能です．

(3) 画面上の決められた位置に，文字を書くときは LOCATE 文を使います．

たとえば，画面上に，縦に「FM7」と書くには，

```
10 CLS
20 LOCATE 10,0:PRINT "F"
30 LOCATE 10,1:PRINT "M"
40 LOCATE 10,2:PRINT "7"
```

これを実行すれば，ディスプレイ上に出力されます．また，：（コロン）を書くことにより複数の命令を同一行に書くことができます．これを，マルチステートメントと呼びます．

(4) 文字に色を付けるときはCOLOR文を使います．

COLOR　　フォアグランドカラー

フォアグランドカラーは，初期状態では下記の色が設定されています．

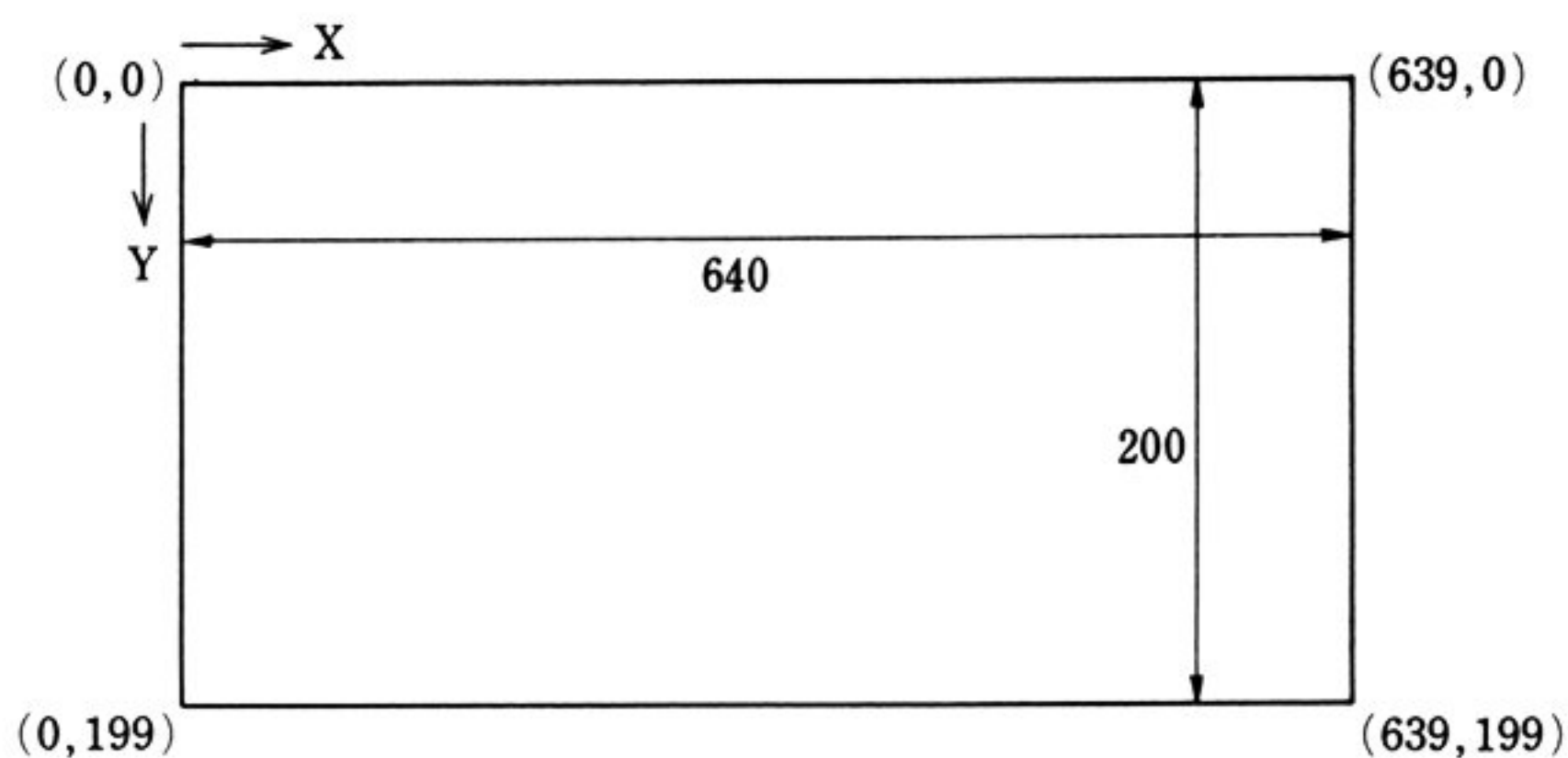
フォアグランドカラー	0	1	2	3	4	5	6	7
色	黒	青	赤	紫	緑	水色	黄	白

```
10 COLOR 4
20 PRINT "FM-7"
30 COLOR 7
```

これを実行すると，ディスプレイ上には，フォアグランドカラー4の緑色で表示されます．

7.2 グラフィック表示

7.2.1 FM-7のグラフィック座標

X
(0,0)
(639,0)
Y
640
200
(0,199)
(639,199)

FM-7のグラフィックは，640×200個のドットについて，それぞれ8色の色指定が可能です．一画面上に，キャラクタ表示と，このグラフィック表示は，同時に混在可能です．

7.2.2 グラフィック表示

グラフィック表示に関するLINE文，CIRCLE文について解説します．その他のグラフィック表示の命令や機能についてはF-BASIC文法書を参照して下さい．

(1) 画面上に線を引くときはLINE文を使います．

LINE $(x_1, \ y_1)-(x_2, \ y_2)$, PSET, パレットコード

x_1, y_1 ⇨ 線の始点座標，x_2, y_2 ⇨ 線の終点座標

ディスプレイ上に対角線を引くには，

```
10 LINE (0,0)-(639,199),PSET,4
20 LINE (639,0)-(0,199),PSET,6
```

このプログラムを実行すると2色に線が引かれます．

(2) 画面上に，円を書くときはCIRCLE文を使います．

CIRCLE (x, y), 半径，パレットコード

(x, y) は中心座標です．では，ディスプレイ上に日の丸を書いてみます．

```
10 CLS
20 LINE(0,0)-(639,199),PSET,7,BF
30 CIRCLE(320,100),130,2
40 PAINT(320,100),2,2
```

・20行の「BF」は，指定された2つの座標を対角線とする4角形を書き，その中を塗りつぶすという指定です．

・40行の「PAINT (320, 100), 2, 2」は，直前で円を書いた中をぬりつぶす命令です．

7.3 単色3画面表示

7.3.1 概　　要

スクリーン機能は，ディスプレイ上に書込む画面，表示する画面を決定するもので，特に，従来できなかった，単色，3画面の表示を可能にしました．

FM-7は，カラー表示のために，3色R(赤)G(緑)B(青) のメモリを持っています．これを，単色として3画面分を別々に利用することが可能です．

7.3.2 スクリーン命令の定義

SCREEN [アクティブVRAMコード][, ディスプレイVRAMコード]

・アクティブVRAMコードは，書込むメモリのコードです．

・ディスプレイVRAMコードは，表示をするメモリのコードです．

・スクリーン命令のコード関係を下に書きます．

緑(G)	赤(R)	青(B)	アクティブVRAMコード	機能	ディスプレイVRAMコード	機能
0	0	0	0	すべてのメモリに書かない	0	すべてのメモリを表示しない
0	0	1	1	Bのメモリのみ書く	1	Bのメモリのみを表示する
0	1	0	2	Rのメモリのみ書く	2	Rのメモリのみを表示する
0	1	1	3	R,B両方に書く	3	R,Bの両方を表示する
1	0	0	4	Gのメモリのみ書く	4	Gのメモリのみを表示する
1	0	1	5	G,B両方に書く	5	G,Bの両方を表示する
1	1	0	6	G,R両方に書く	6	G,Rの両方を表示する
1	1	1	7	G,R,Bすべてに書く	7	G,R,Bすべてを表示する

初期状態では，アクティブVRAMコード＝7，ディスプレイVRAMコード＝7 に設定されており，その意味はすべてのメモリに書き，すべてのメモリを表示するというものです．

初期状態で，画面をクリアしてから次のプログラムを実行してみます．

```
10 COLOR 7,0
20 SCREEN 7,7
30 PRINT "ABCD"
```

画面に，白でABCDと表示されます．

では，本当に画面が3枚のRGBのメモリにより表示されているのなら，これらをばらばらに見ることが可能なはずです．実際に，それを見てみます．

SCREEN 7, 1

と入力すると，青のメモリの様子が見えるはずです．画面には青で表示されます．青のメモリに書かれた様子がわかります．

同様にほかのメモリも，ディスプレイVRAMコードをさきほどの表に従って入力すれば確認できます．

(例) SCREEN 7, 2 ← 赤のメモリの様子を示す．

(1) アクティブVRAMコード，ディスプレイVRAMコードについて

SCREEN文でアクティブVRAMコード，ディスプレイVRAMコードを設定した後は，画面表示コマンド（CLS文，PRINT文，LINE文，CIRCLE文等）の実行結果がアクティブVRAMに書き込まれ，ディスプレイVRAMの内容が画面に表示されます．

・アクティブVRAMコード――画面上のデータを書き込むVRAM（B，R，G）を指定

・ディスプレイVRAMコード――画面上に表示するVRAM（B，R，G）を指定

VRAM（B，R，G）はサブCPU上にあり，メインCPUとVRAMとのデータの読み込み，書き込み制御は，このSCREEN文によって行います．以下に，プログラム例を使用して説明します．

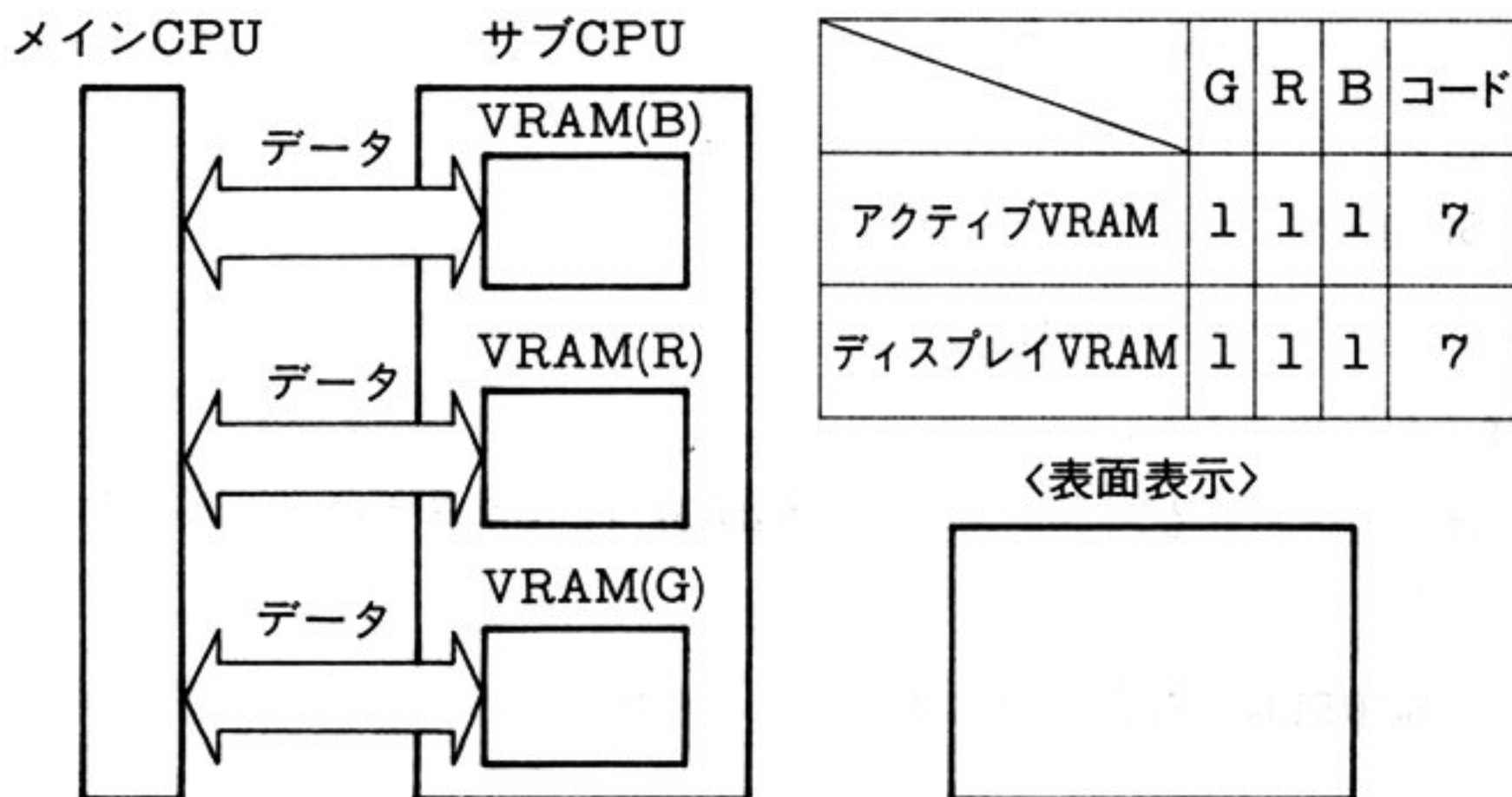
〈プログラム例〉

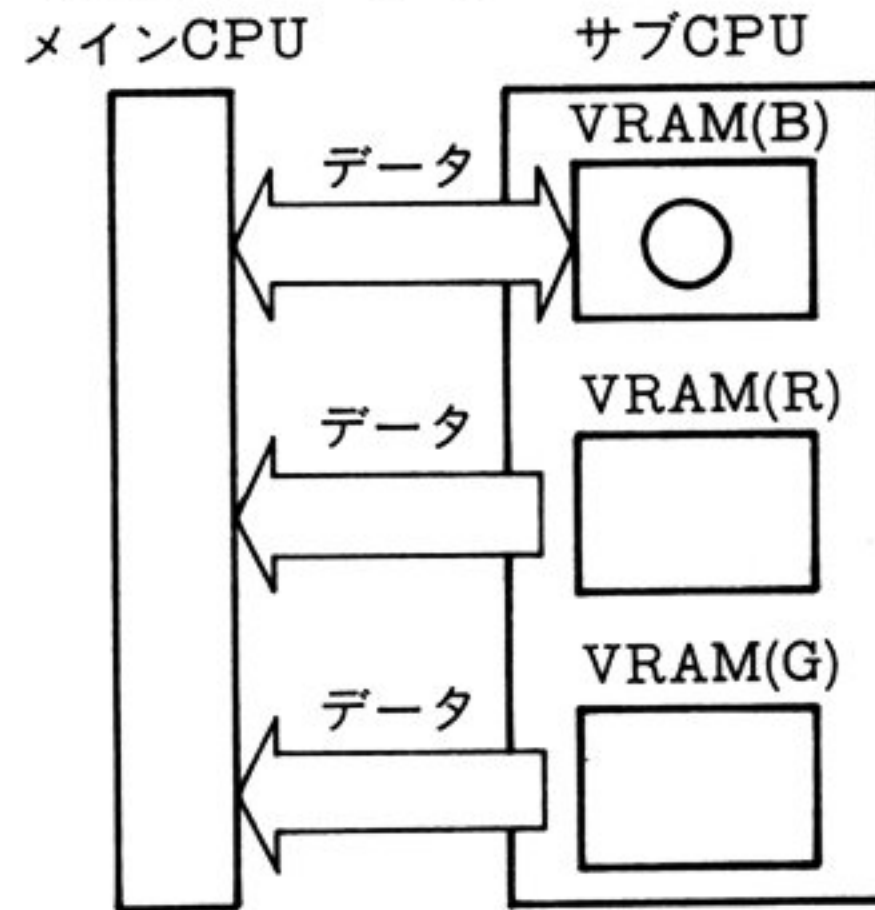
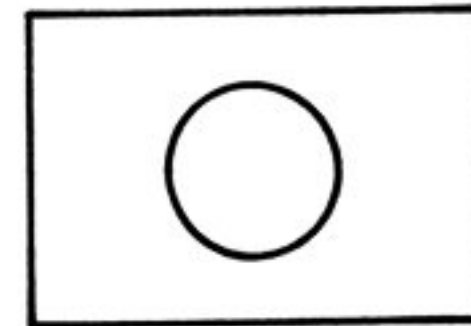
```
10 SCREEN 7, 7
20 CLS
30 SCREEN 1, 7
40 CIRCLE(320, 100), 50, 1
50 SCREEN 2, 7
60 LINE(253, 70)-(387, 130), PSET, 2, B
70 SCREEN 4, 7
80 CONNECT(320, 81)-(353, 111)-(287, 111)-(320, 81), 4
90 SCREEN 1, 0
100 LOCATE 0, 0:PRINT "B"
110 SCREEN 2, 0
120 LOCATE 0, 9:PRINT "R"
130 SCREEN 4, 0
140 LOCATE 0, 19:PRINT "G"
150 SCREEN 0, 5
160 SCREEN 0, 6
```

〈実行過程〉（図中の矢印はデータの流れる方向を示します．）

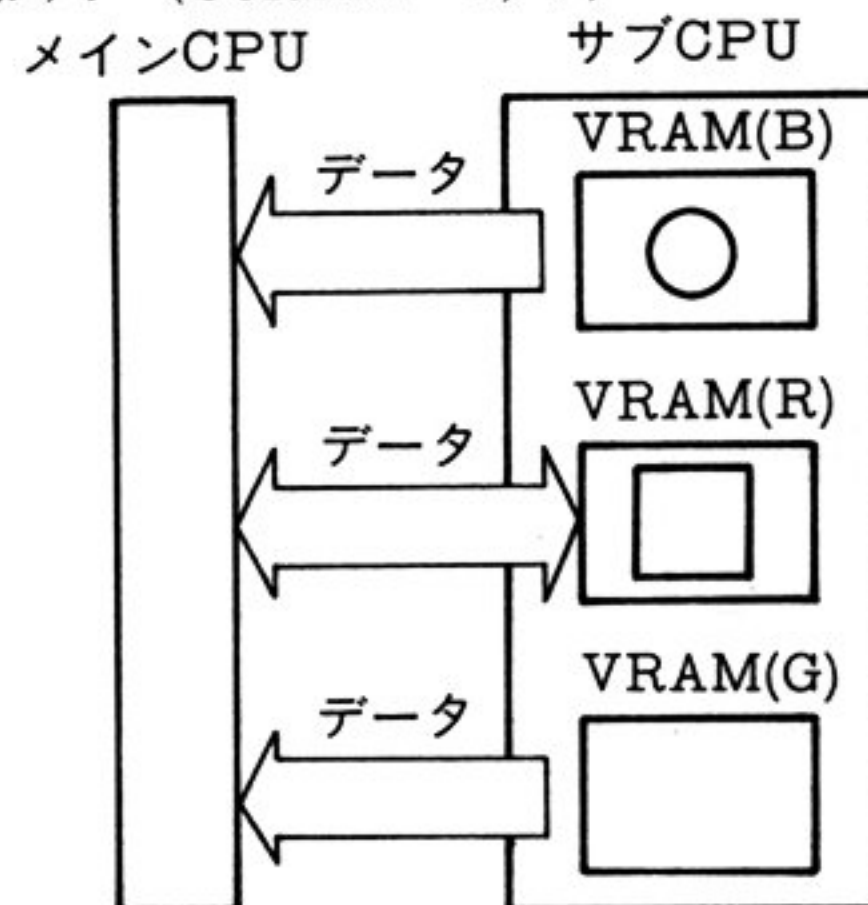
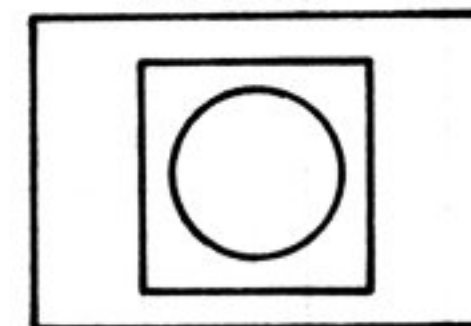
・10−20行　アクティブVRAMをB，R，G，ディスプレイVRAMをB，R，Gに設定して画面をクリアします．（SCREEN 7，7）

	G	R	B	コード
アクティブVRAM	1	1	1	7
ディスプレイVRAM	1	1	1	7

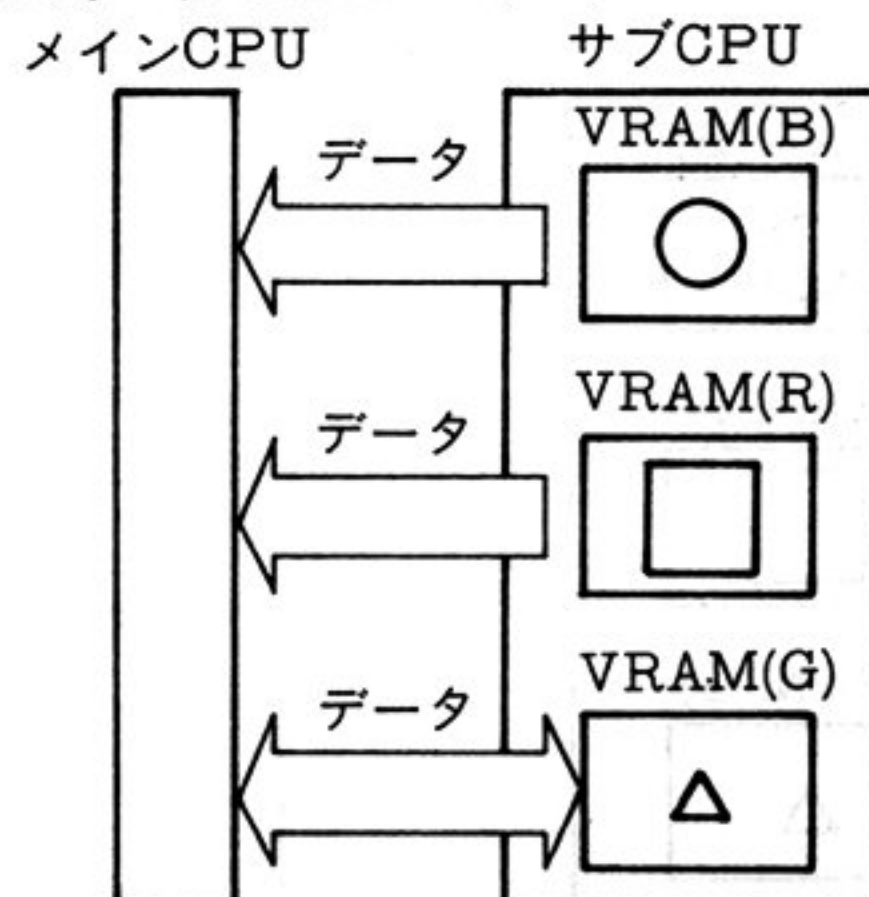
〈表面表示〉

・30－40行　アクティブVRAMをB，ディスプレイ VRAMをB，R，Gに設定して画面上に円を描きます．（SCREEN　1，7）

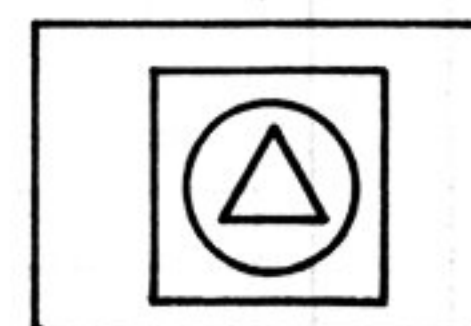
メインCPU　サブCPU
VRAM(B)　VRAM(R)　VRAM(G)
データ　データ　データ

	G	R	B	コード
アクティブVRAM	0	0	1	1
ディスプレイVRAM	1	1	1	7

〈画面表示〉

・50－60行　アクティブVRAMをR，ディスプレイ VRAMをB，R，Gに設定して画面上に四角形を描きます．（SCREEN　2，7）

メインCPU　サブCPU
VRAM(B)　VRAM(R)　VRAM(G)
データ　データ　データ

	G	R	B	コード
アクティブVRAM	0	1	0	2
ディスプレイVRAM	1	1	1	7

〈画面表示〉

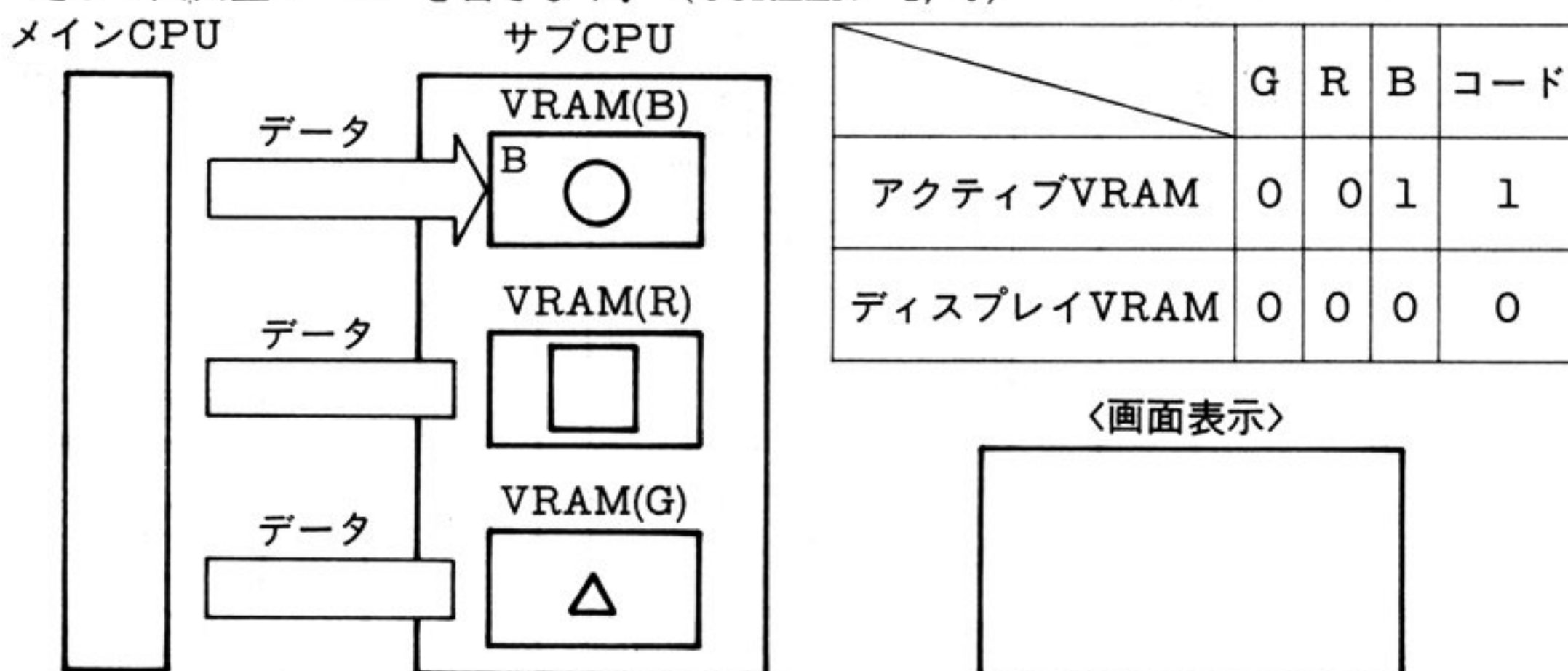
・70－80行　アクティブVRAMをG，ディスプレイ VRAMをB，R，Gに設定して画面上に三角形を描きます．（SCREEN　4，7）

メインCPU　サブCPU
VRAM(B)　VRAM(R)　VRAM(G)
データ　データ　データ

	G	R	B	コード
アクティブVRAM	1	0	0	4
ディスプレイVRAM	1	1	1	7

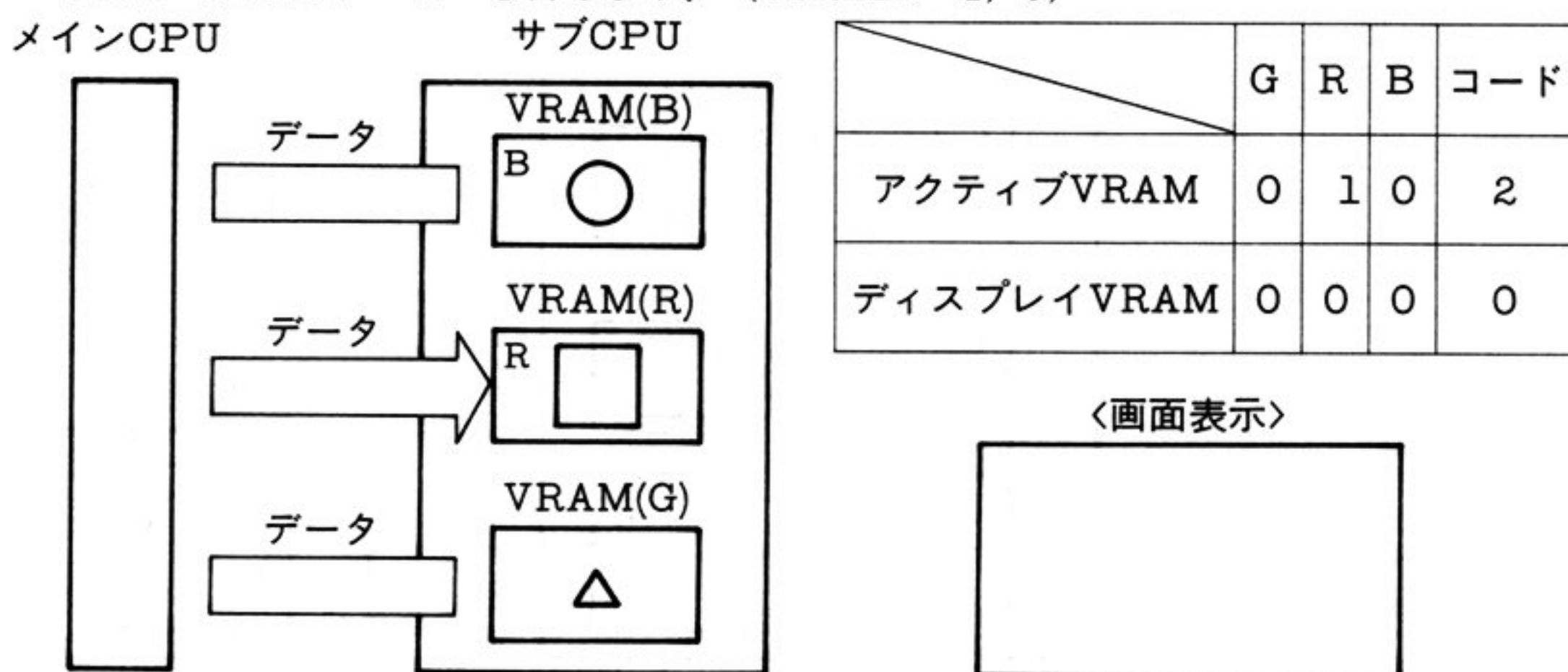
〈画面表示〉

・90－100行　アクティブVRAMをB，全てのディスプレイVRAM（B，R，G）の表示を不可能に設定して画面上に“B”を書きます．（SCREEN　1，0）

メインCPU　サブCPU
データ　VRAM(B)　B
データ　VRAM(R)
データ　VRAM(G)

	G	R	B	コード
アクティブVRAM	0	0	1	1
ディスプレイVRAM	0	0	0	0

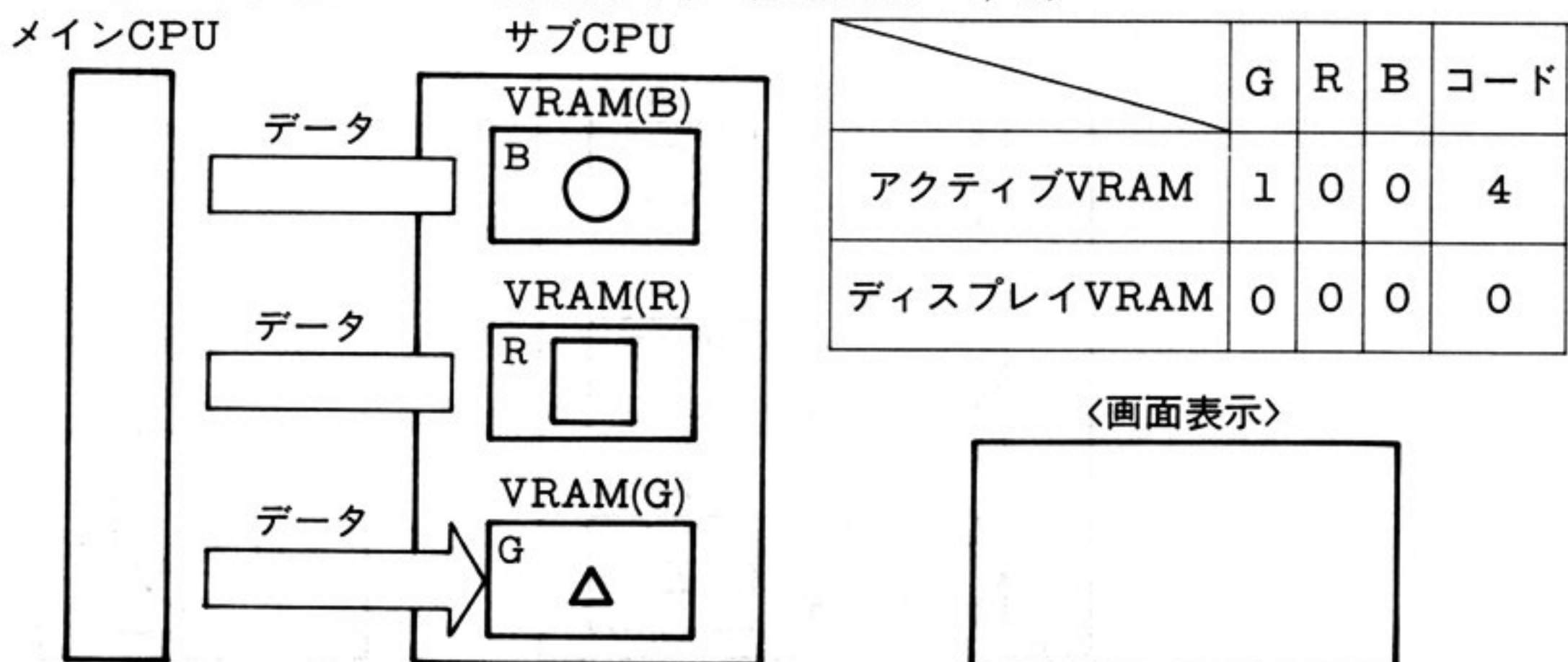
〈画面表示〉

・110－120行　アクティブVRAMをR，全てのディスプレイVRAM（B，R，G）の表示を不可能に設定して画面上に“R”を書きます．（SCREEN　2，0）

メインCPU　サブCPU
データ　VRAM(B)　B
データ　VRAM(R)　R
データ　VRAM(G)

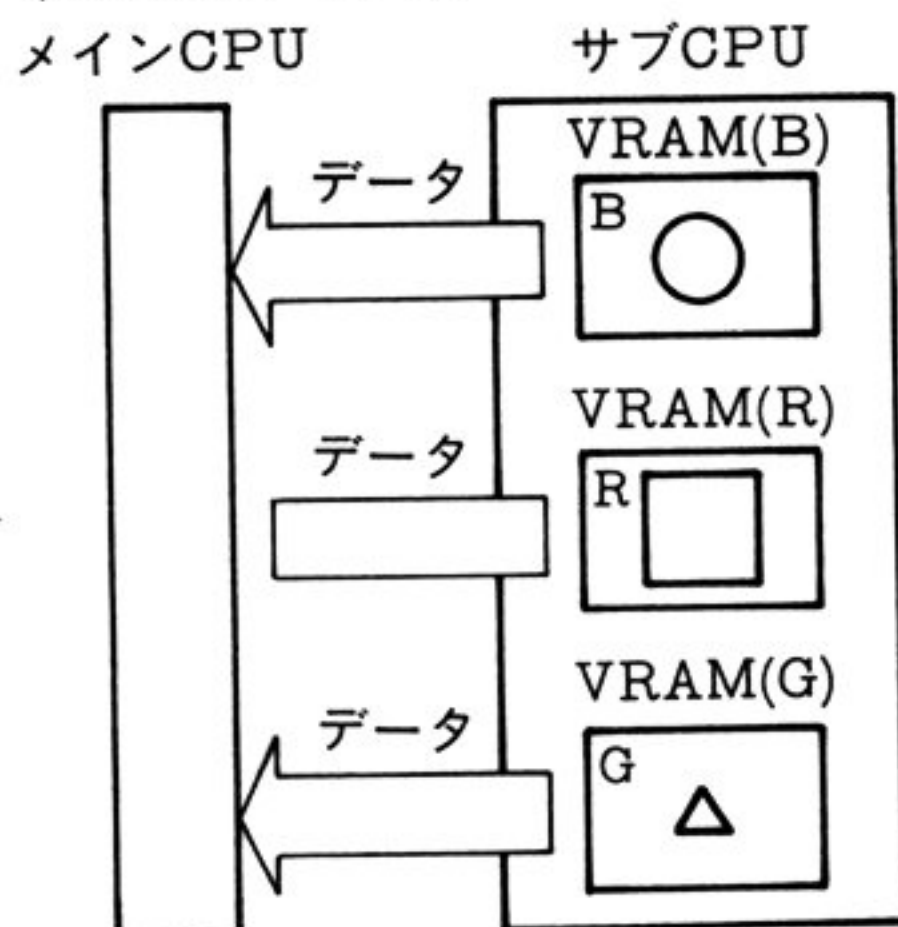
	G	R	B	コード
アクティブVRAM	0	1	0	2
ディスプレイVRAM	0	0	0	0

〈画面表示〉

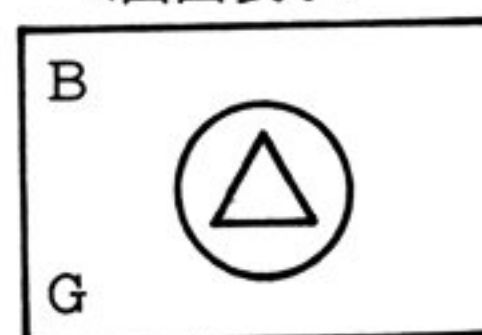
・130－140行　アクティブVRAMをG，全てのディスプレイVRAM（B，R，G）の表示を不可能に設定して画面上に“G”を書きます．（SCREEN　4，0）

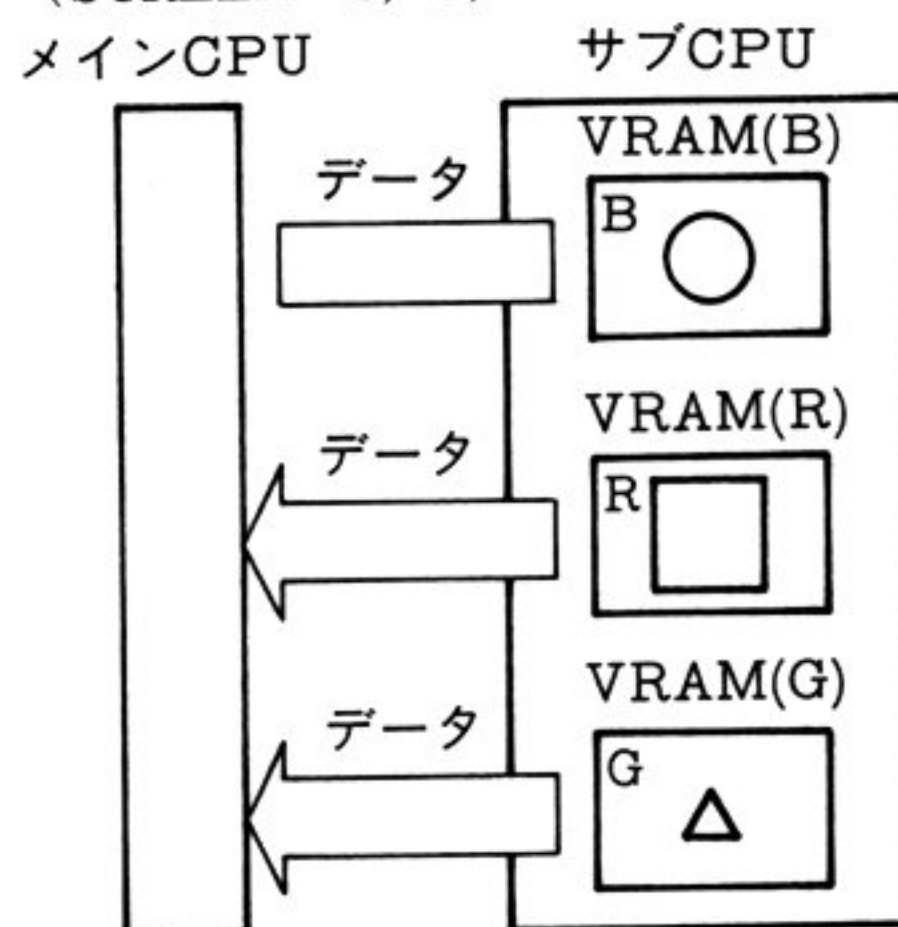
メインCPU　サブCPU
データ　VRAM(B)　B
データ　VRAM(R)　R
データ　VRAM(G)　G

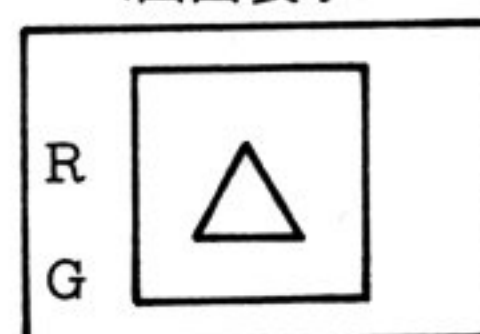
	G	R	B	コード
アクティブVRAM	1	0	0	4
ディスプレイVRAM	0	0	0	0

〈画面表示〉

・150行　全てのVRAM（B, R, G）への書き込みを不可能にし，ディスプレイ VRAM をB, Gに設定します．（SCREEN　0, 5）

メインCPU　サブCPU
VRAM(B)　B　データ
VRAM(R)　R　データ
VRAM(G)　G　データ

	G	R	B	コード
アクティブVRAM	0	0	0	0
ディスプレイVRAM	1	0	1	5

〈画面表示〉
B
G

・160行　全ての VRAM（B, R, G）への書き込みを不可能にし，ディスプレイ VRAM をR, Gに設定します．（SCREEN　0, 6）

メインCPU　サブCPU
VRAM(B)　B　データ
VRAM(R)　R　データ
VRAM(G)　G　データ

	G	R	B	コード
アクティブVRAM	0	0	0	0
ディスプレイVRAM	1	1	0	6

〈画面表示〉
R
G

次頁より，アクティブでない VRAM（上の説明中のアクティブ VRAM が“0”になっている VRAM）をノンアクティブ VRAM と呼びます．

（2） スクリーン命令使用上の注意事項

スクリーン命令でアクティブ VRAM コードを 7 以外に設定して PAINT 文，GET@（形式 3）文，GET@A（形式 4）文を使用する際の注意事項を以下に述べます．

SCREEN 文でアクティブ VRAM コードを 1（青）に設定した場合に，パレット番号 1（青），3（紫），5（水色），7（白）でグラフィック画面を描いた時，グラフィックの描かれたドットは次の表のように VRAM の G，R，B に書き込まれます．

緑 （G）	赤 （R）	青 （B）
※	※	1

※・・・・・以前の値（0　or　1）を保持

ノンアクティブVRAM（アクティブでないVRAM）の内容は，SCREEN 文を定義する前の値を保持します．

この後に，PAINT 文，GET@（形式 3）文，GET@A（形式 4）文を実行すると，アクティブVRAM だけでなく，ノンアクティブ VRAM の内容も参照します．この際，ノンアクティブ VRAM の内容は，不定の値となります．この状態で考えられるノンアクティブ VRAM の内容は，次の 4 通りあります．

緑 （G）	赤 （R）	青 （B）	〔パレットコード〕
0	0	1	・・・・・・・1（青）
0	1		・・・・・・・3（紫）
1	0		・・・・・・・5（水色）
1	1		・・・・・・・7（白）

従って，PAINT 文における境界色，GET@（形式３）文におけるパレットコードを指定する時に，この全ての場合を考慮する必要があります．

以下に，各コマンドごとの説明を述べます．

（ａ） PAINT 文を実行する場合

```
10 SCREEN 7, 7
20 CLS
30 SCREEN 1, 7
40 CIRCLE (320, 100), 50, 5
50 PAINT (320, 100), 7, 5
```

上記のプログラムを実行した場合の実行過程は次のようになります．

・10－20行でＧ，Ｒ，ＢのＶＲＡＭの全てのVRAM をクリアします．

・30行でアクティブ VRAM コードを１(青)に指定します．つまり，Ｂがアクティブで，ＧとＲがノンアクティブとなります．

・40行で半径50ドットの円をＧとＢの VRAM に書き込もうとしますが，Ｇはノンアクティブになっているので，Ｂだけに円を書き込みます（画面上では青色の円が表示されます）。

・50行で円の内部を白色で塗りつぶします．

50行を実行する時に，Ｇ，Ｒ，Ｂの VRAM の内容を調べて塗りつぶすかどうかを判断しますが，先に述べたようにノンアクティブ VRAM であるＧとＢの内容は不定の値となり，そのため，境界となるはずのドットのパレットコードが境界色で指定したパレットコードと一致しなくなります．この場合は円の外部も塗りつぶされることになります．

従って，ノンアクティブ VRAM の内容を考慮して境界色を指定するために，上記のプログラムは次のように書き換えます．

```
10 SCREEN 7, 7
20 CLS
30 SCREEN 1, 7
40 CIRCLE (320, 100), 50, 5
50 PAINT (320, 100), 7, 1, 3, 5, 7
```

（1, 3, 5, 7 → ノンアクティブ VRAM の内容を全て考慮します．）

アクティブ VRAM コードと，その時の PAINT 文で指定すべき境界色との関係を以下の表に示します．

SCREEN 文の アクティブ VRAM コード	PAINT 文で指定すべき境界色
1	1, 3, 5, 7
2	2, 3, 6, 7
3	1 を境界色としたい場合・・・・・・・1, 5 2 を境界色としたい場合・・・・・・・2, 6 3 を境界色としたい場合・・・・・・・3, 7 5 を境界色としたい場合・・・・・・・1, 5 6 を境界色としたい場合・・・・・・・2, 6 7 を境界色としたい場合・・・・・・・3, 7
4	4, 5, 6, 7
5	1 を境界色としたい場合・・・・・・・1, 3 3 を境界色としたい場合・・・・・・・1, 3 4 を境界色としたい場合・・・・・・・4, 6 5 を境界色としたい場合・・・・・・・5, 7 6 を境界色としたい場合・・・・・・・4, 6 7 を境界色としたい場合・・・・・・・5, 7
6	2 を境界色としたい場合・・・・・・・2, 3 3 を境界色としたい場合・・・・・・・2, 3 4 を境界色としたい場合・・・・・・・4, 5 5 を境界色としたい場合・・・・・・・4, 5 6 を境界色としたい場合・・・・・・・6, 7 7 を境界色としたい場合・・・・・・・6, 7

（ｂ）　GET@（形式３）文を実行する場合

GETするドットを配列に取り込む際に，ノンアクティブVRAMの内容が不定であるのでGETするドットのパレットコードを複数個指定します．アクティブVRAMコードと，その時のGET@文で指定すべきパレットコードとの関係は（ａ）と同じです．

```
10 CLS
20 DIM  A%(350)
30 SCREEN  2, 7
40 LINE (100, 50)-(200, 100), PSET, 7, BF
50 GET@ (100, 50), A%, G, 2
60 PUT@ (200, 100)-(300, 150), A%, PSET, 7
```

上記のプログラムは次のように書き換えます．

```
10 CLS
20 DIM  A%(350)
30 SCREEN  2, 7
40 LINE (100, 50)-(200, 100), PSET, 7, BF
50 GET@ (100, 50)-(200, 100), A%, G, 2, 3, 6, 7
60 PUT@ (200, 100)-(300, 150), A%, PSET, 7
```

ノンアクティブVRAMの内容を全て考慮します．

（ｃ）　GET@A（形式４）文を実行する場合

この場合は，GETするドットのG，R，B全てのVRAMの内容を配列に取り組み，その際の，ノンアクティブVRAMの内容は不定な値となります．

```
10 CLS
20 DIM  A%(1000)
30 SCREEN  3, 7
40 LINE (100, 50)-(200, 100), PSET, 7, BF
50 GET@A (100, 50)-(200, 100), A%, G
60 PUT@A (200, 100)-(300, 150), A%, PSET
```

上記のプログラムでは，GETする時とPUTする時のSCREEN文が同じなので，GETした時のアクティブVRAMのデータを同じアクティブVRAMにPUTするので同じ画面が表示されます．このプログラム中の55行にSCREEN文でアクティブVRAMコードを変更してPUTを行うと，GETしたものと違う画面が表示されます．従って，このコマンドを使用する際には，SCREEN文のアクティブVRAMコードを７にすることが望ましいです．

7.4 パレット機能

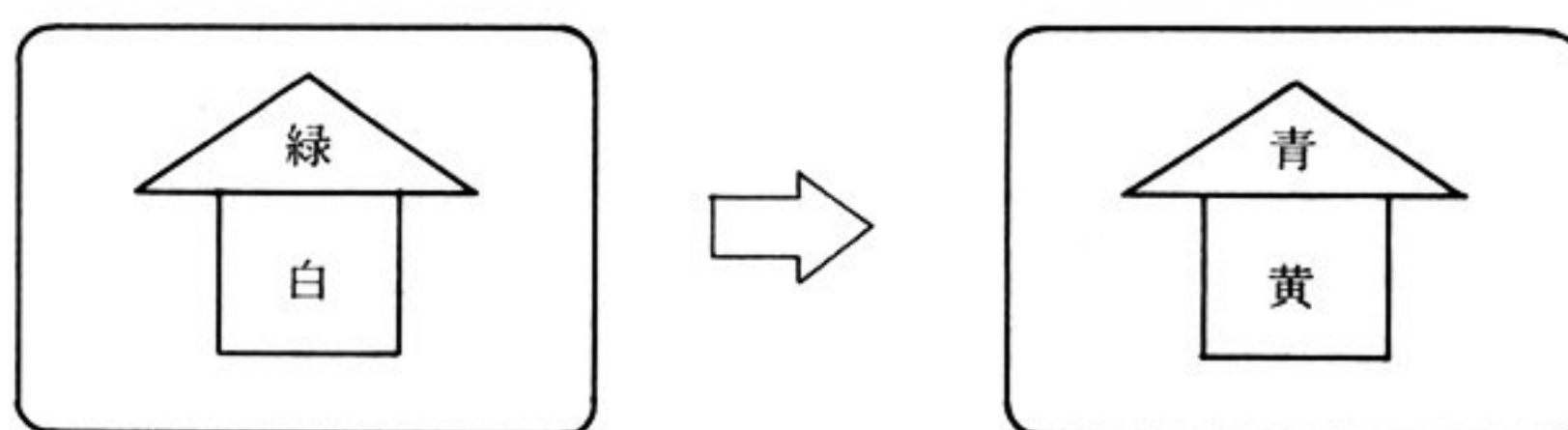
7.4.1 概　　要

パレット機能は，8色の色の対応を決める機能です．

カラーコードでは0－7までそれぞれ固有の色番号を持っていますが，この色に，ユーザは自由に番号を設定して，独自の番号に変更することができます．これがパレット機能です．

このパレット機能は，プログラム中で自由に使用できます．

また，このパレット機能は，ディスプレイ上のすべての色に対して作用します．

この機能を使用しますと，たとえば現在ディスプレイ上に表示されている絵の色を，絵の書き換えをせずに変更することが可能です．

7.4.2 パレットコードの定義

パレットコードとカラーコードの対応を定義します．

COLOR＝(パレットコード，カラーコード)

BASICが起動されたときのパレットコードとカラーコードの対応は次のように設定されています．

カラーコード表

0	1	2	3	4	5	6	7
黒	青	赤	紫	緑	水色	黄	白

パレットコード表

パレットコード／状態	0	1	2	3	4	5	6	7
初期状態	黒	青	赤	紫	緑	水色	黄	白
パレット指定時	0 黒	0～7 黒～白	0～7 黒～白	0～7 黒～白	0～7 黒～白	0～7 黒～白	0～7 黒～白	0～7 黒～白

たとえば，初期状態で

```
20 COLOR 2
30 PRINT "ABCD"
40 COLOR 7
50 END
```

このプログラムを実行した場合，「20　COLOR　2」の文で，2はパレットコードの2番を意味し，同時にカラーコード2の赤に対応しており，ディスプレイ上には赤でABCDと表示されることとなります．

次にパレットコードの指定をします．

```
10 COLOR=(2,6)
20 COLOR 2
30 PRINT "ABCD"
40 COLOR 7
50 END
```

これは「10　COLOR＝(2, 6)」により，パレットコード2はカラーコード6黄色として扱えということになり，ディスプレイ上には黄色でABCDと表示されます．また同様にパレットコード6で書いた文字も黄色で出ます．また，これは次にパレット指定するまで変更されません．

パレットコードの指定を初期状態にするには，次のようにします．

```
10 FOR I=0 TO 7
20 COLOR=(I,I)
30 NEXT
40 END
```

7.4.3　COLOR文の動作

COLOR文では実際にどんな動作をするか説明します．

```
10 CLS
20 FOR I=0 TO 7
30 COLOR=(I,I)
40 NEXT
50 FOR I=0 TO 7
60 COLOR I
70 PRINT "PALETTE = ";I
80 NEXT
90 END
```

このプログラムを実行すると，ディスプレイ上に，8色（黒色も含む）の文字が現われます．

そこで，直接モードで次の命令を実行して下さい．

```
COLOR =(3, 5)
```

同時に今まで紫色で書いてあった文字が，パレットコードの指定により水色に変更されます．元にもどすことは，直接モードで次の命令を実行します．

COLOR＝(3, 3)

これでもとの紫色にもどります.

7.5 プログラム例

(1) プログラム (例1)

LINE 文を使ってディスプレイ上に 64 色の色を出力するプログラムです.

```
10 CLS
20 FOR I=0 TO 7
30 FOR J=0 TO 7
40 FOR K=0 TO 9 STEP 2
50 LINE(K+J*10+I*80,50)-(K+J*10+I*80,150),PSET,I
60 LINE(K+J*10+1+I*80,50)-(K+J*10+1+I*80,150),PSET,J
70 NEXT K
80 NEXT J
90 NEXT I
```

(2) プログラム (例2)

スクリーン命令, カラー命令等を使って, 単色, 3 画面を表示するプログラムです.

```
10 SCREEN 7,7:CLS
20 COLOR=(1,7)
30 COLOR=(2,7)
40 COLOR=(4,7)
50 SCREEN 1,1:LOCATE 5,0:PRINT "*** SCREEN 1 ***"
60 SCREEN 2,2:LOCATE 5,0:PRINT "CCC SCREEN 2 ゴゴゴ"
70 SCREEN 4,4:LOCATE 5,0:PRINT "+++ SCREEN 3 +++"
80 FOR I=0 TO 2
90 SCREEN 2^I,2^I
100 LOCATE 0,5:PRINT "NO,";I+1;"MESSAGE = ";:LINE INPUT A$
110 NEXT I
120 FOR I=0 TO 2
130 SCREEN 2^I,2^I
140 GOSUB 160
150 NEXT I:GO TO 120
160 FOR G=1 TO 1000:NEXT G:IF INKEY$=" " THEN 180
170 RETURN
180 FOR I=0 TO 7:COLOR=(I,I):NEXT
190 SCREEN 7,7:CLS
200 END
```

このプログラムを実行すると, 画面上に SCREEN 1 ～ 3 まで表示したい文字を入力します. 入力終了は ⏎ キーで行ないます. 3 画面分入力すると, 自動的に 3 画面を次々と表示します. 終了はスペースキーで行ないます.

(3) プログラム(例3)

プログラム(例1)に，スクリーン命令等を追加し，64色の色がさらに順次変わるプログラムです.

```
10 CLS
20 FOR I=0 TO 7:COLOR=(I,I):NEXT I
30 FOR I=0 TO 7
40 FOR J=0 TO 7
50 FOR K=0 TO 9 STEP 2
60 LINE (K+J*10+I*80,50)-(K+J*10+I*80,150),PSET,I
70 LINE (1+K+J*10+I*80,50)-(1+K+J*10+I*80,150),PSET,J
80 NEXT K
90 NEXT J
100 NEXT I
110 FOR J=1 TO 7
120 COLOR=(J,(J+L) MOD 7)
130 NEXT J:L=L+1:IF L=7 THEN L=0
140 GOSUB 160
150 GOTO 110
160 FOR K=0 TO 1000:NEXT K:IF INKEY$=" " THEN 180
170 RETURN
180 FOR I=0 TO 7:COLOR=(I,I):NEXT
190 END
```

第8章　音楽演奏機能

8.1 概　　要

FM-7は音楽演奏用LSIとしてGI社のProgramable Sound Generatorを実装しており，3重和音での音楽演奏機能があります．F-BASIC V3.0にはPLAY文とSOUND文という2個の命令があります．PLAY文はMusic Macro Languageを使用して容易に音楽演奏が可能であります．また，SOUND文はProgramable Sound Generatorを直接制御し，ゲームの効果音たとえば波の音，飛行機の墜落音などを容易に出力可能とします．

これらの命令を使用することにより，美しい音楽や迫力あるゲーム楽しみながらFM-7で音の世界を広げることができます．

8.2 PLAY文の使い方

PLAYのMML(Music Macro Language)を使用して音楽演奏を行なうことができます．このMMLはパソコンで音楽演奏を容易に行なうための言語です．このMMLにはテンポ，音程，音長，波形などを指定するコマンドがあり，これらのコマンドを使用して音楽を演奏できます．以下にそれぞれのコマンドについて説明します．

8.2.1 音　量(Vn)

このコマンドは音量を指定します．nの値は整数で0から15まで使用できます．ここで15は音量が最大で，0ではそのパートの音が出力されません．使用例は次のようになります．

```
10 PLAY "V15"
          ↑
          └─音量を最大に指定
```

8.2.2 テンポ(Tn)

このコマンドは音を出す速度を指定します．nの値は32から255までの整数を使用します．Tnの速度は1分間に4分音符をn回数える速さであります．

使用例は次のようになります．

```
20 PLAY "T100"
```
（T100 ← テンポを♩＝100に指定）

最初の Tn の値は 100 に設定してあります．

8.2.3 休　符 (Rn)

このコマンドは休符を表わし，n は 1 から 64 までの整数を使用します．休符の指定例を次の表に示します．

休　符　の　指　定

休符	名称	指定	休符	名称	指定
	全　休　符	R1		付点 16 分休符	R16R32
	付点 2 分休符	R2R4		16 分 休 符	R16
	2 分 休 符	R2			R32R32
	付点 4 分休符	R4R8		32 分 休 符	R32
	4 分 休 符	R4			R64R64
	付点 8 分休符	R8R16		64 分 休 符	R64
	8 分 休 符	R8			

8.2.4 音　程 (A, B, C, D, E, F, G)

この A から G までのコマンドは音程を表わします．C＝ド，D＝レ，E＝ミ，F＝ファ，G＝ソ，A＝ラ，B＝シに対応しています．また，このコマンドのうしろに #，＋(#)，－(b) を加えて半音を表わすことができます．

8.2.5 オクターブ (On)

このコマンドは，音程のオクターブを指定します．n は 1 から 8 までの整数を使用します．なお基準周波数の 440Hz の音はオクターブ 4 (O4) のラの音 (A) となります．なお，最初の On の値は 4 に設定してあります．

8.2.6 音　長 (Ln)

このコマンドは音の長さを指定します．n は 1 から 64 の値で表わし，n の値が 1 の時最長で値の逆数が長さとなります．なお，最初の設定は 4 です．

次の表に音符と音長の指定例を示します．

音 符 の 指 定

𝅝	全 音 符	L1	L1A	𝅘𝅥𝅯.	付点16分音符	—	L16A.
𝅗𝅥.	付点2分音符	—	L2A.	𝅘𝅥𝅯	16 分 音 符	L16	L16A
𝅗𝅥	2 分 音 符	L2	L2A	𝅘𝅥𝅰.	付点32分音符	—	L32A.
♩.	付点4分音符	—	L4A.	𝅘𝅥𝅰	32 分 音 符	L32	L32A
♩	4 分 音 符	L4	L4A	𝅘𝅥𝅱.	付点64分音符	—	L64A.
♪.	付点8分音符	—	L8A.	𝅘𝅥𝅱	64 分 音 符	L64	L64A
♪	8 分 音 符	L8	L8A				

次に5線符の音をプログラムで作成してみます.

♩=120

```
10  PLAY "V15"
20  PLAY "T120"
30  PLAY "L4O4CDEFGABO5CR1"
```

- R1：全休符を指定
- O5：5オクターブのドを指定
- O4：以下の音程を4オクターブに指定
- L4：以下の音長を4（4分音符）に指定

8.2.7 特殊音程（Nn）

このコマンドは音程の特殊形で, nの値は1から96までの整数で表します. N1はO1Cを示し, N96はO8Bまで指定できます.

音程と特殊音程の対応表を以下に示します.

音程と特殊音程（Nn）の対応表

音程 ＼ オクターブ		O1	O2	O3	O4	O5	O6	O7	O8
シ	B	N12	N24	N36	N48	N60	N72	N84	N96
	A♯ A＋ B－	N11	N23	N35	N47	N59	N71	N83	N95
ラ	A	N10	N22	N34	N46	N58	N70	N82	N94
	G♯ G＋ A－	N9	N21	N33	N45	N57	N69	N81	N93
ソ	G	N8	N20	N32	N44	N56	N68	N80	N92
	F♯ F＋ G－	N7	N19	N31	N43	N55	N67	N79	N91
ファ	F	N6	N18	N30	N42	N54	N66	N78	N90
ミ	E	N5	N17	N29	N41	N53	N65	N77	N89
	D♯ D＋ E－	N4	N16	N28	N40	N52	N64	N76	N88

音程と特殊音程(Nn)の対応表のつづき

音程 ＼ オクターブ	O1	O2	O3	O4	O5	O6	O7	O8
レ D	N3	N15	N27	N39	N51	N63	N75	N87
C# C+ D−	N2	N14	N26	N38	N50	N62	N74	N86
ド C	N1	N13	N25	N37	N49	N61	N73	N85

N46 = O4A = 440Hz

次のような5線符の音をプログラムで作成します.

```
10  PLAY "V5"
20  PLAY "T120L4"
30  FOR I=49 TO 37 STEP -1
40  PLAY "N=I;"
50  NEXT
```

次はリピート・アンド・フェード・アウト(Rep. & F.O)の5線符をプログラムで作成します.

Rep. & F.O.

```
10  PLAY "T120L4O5"
20  FOR I=12 TO 2 STEP -1
30  PLAY "V=I; CRCR"
40  NEXT I
```

8.2.8 エンベロープパターン(Sn)

このコマンドは, エンベロープを指定します. nは0から15までの整数を使用します. このコマンドを使用することにより, Vコマンドは無効となります. Snコマンドの解除は, 再びVコマンドで音量を指定することにより実行されます.

8.2.9 エンベロープ周波数(Mn)

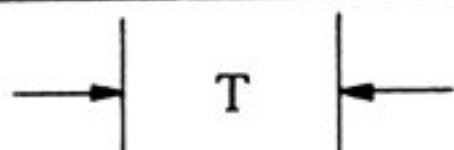
このコマンドは, Sコマンドとペアで使用します. nは0から65535までの整数を使用します. エンベロープパターンと周波数の計算式を下に示します.

$$T = \frac{256 \times n}{f} \quad (秒) \qquad n = 1 \sim 65535$$

$$= \frac{2.08 \times n}{10^4} \quad ただし, f = 1.2288\text{MHz}$$

エンベロープパターンとSコマンドの数値nとの対応表

S_n	エンベロープパターン
0～3,9	
4～7,15	
8	
10	
11	
12	
13	
14	

T

以下のような5線符をこれまでの知識でプログラムを作成するとCの音が連続して聞えます．そこでエンベロープを使用して新しいプログラムを作成します．

♩=120

```
10  PLAY "V8T120L4"
20  PLAY "S13M50"
30  PLAY "O4CDEFGABO5CCO4BAGFEDC"
40  GOTO 30
```

8.3　PLAY 文による和音

FM-7が使用しているPSG(Programmable Sound Generator)は3個の音源を持っているため，3重和音を出すことができます．

PLAY文を使用して，次のようなコードをプログラムで作成してみます．

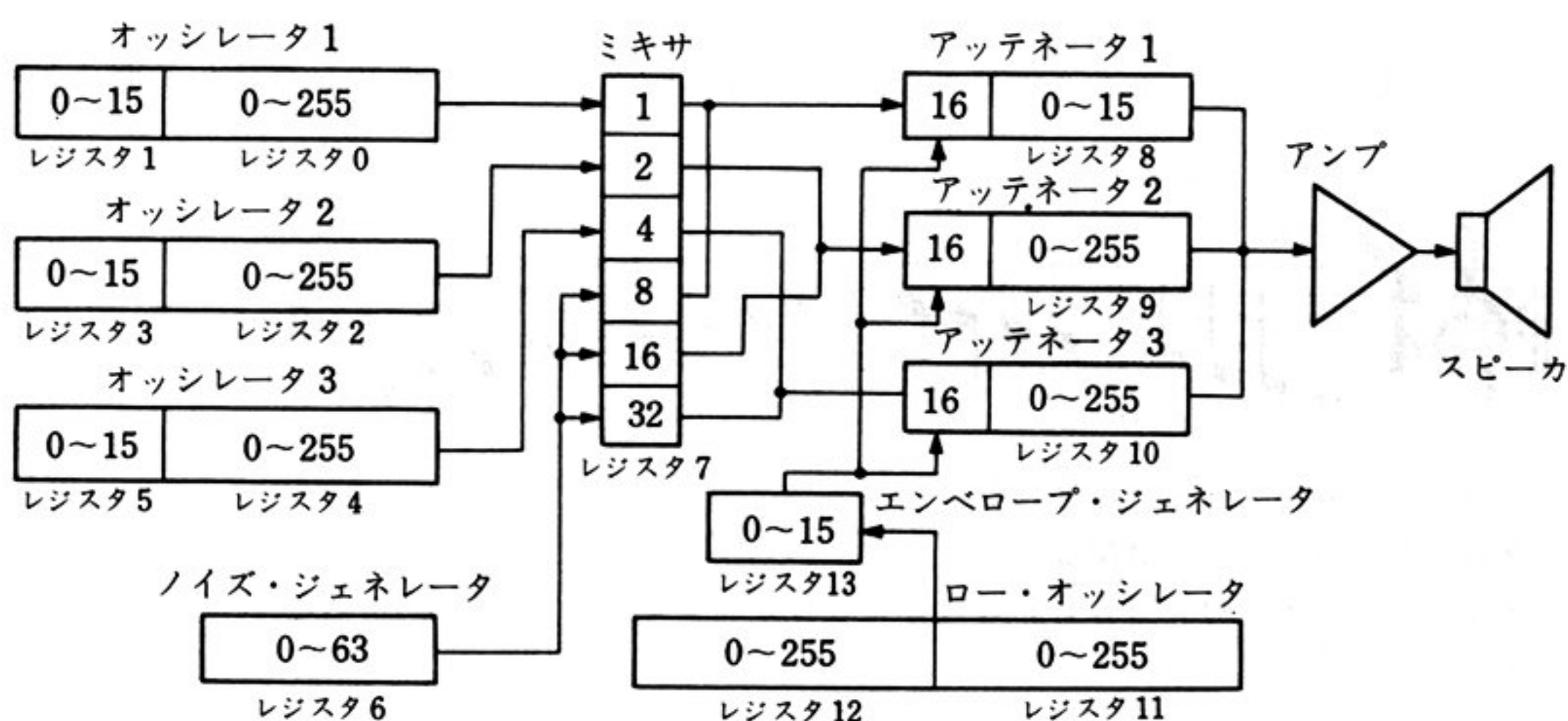
C　G_7　F

```
10  PLAY "T120L4", "T120L4", "T120L4"
20  PLAY "S9M2000", "S9M2000", "S9M2000"
30  FOR  I=1  TO  5
40  PLAY "04CGGG","R05CCC","R05EEE"
50  PLAY "03G04GGG", "R04BBB", "R05FFF"
60  PLAY "04FAAA", "R05CCC", "R05FFF"
70  NEXT  I
```

8.4 SOUND文の使い方

SOUND文は，FM-7が使用しているPSG(Programable Sound Generator)を直接制御することとができます．

8.4.1 PSGの概要

FM-7に使用しているPSGは，下図に示すようにオシレータ部，ミキサ部，アッテネータ部，ノイズ発生部，エンベロープ発生部およびミキシングアンプ部から構成されます．

PSGのブロック

PSGの各ブロックは，14個のレジスタが対応しており，それぞれのレジスタの機能を次の表に示します．

レジスタ ＼ ビット	働き	7	6	5	4	3	2	1	0
R_0	チャンネルAの音階	下位8ビット・データ							
R_1						上位4ビット・データ			
R_2	チャンネルBの音階	下位8ビット・データ							
R_3						上位4ビット・データ			
R_4	チャンネルCの音階	下位8ビット・データ							
R_5						上位4ビット・データ			
R_6	ノイズ周波数	6ビット・データ							
R_7	入出力の選択			ノイズ C	ノイズ B	ノイズ A	トーン C	トーン B	トーン A
R_8	チャンネルAの音量					4ビット・データ			
R_9	チャンネルBの音量					4ビット・データ			
R_{10}	チャンネルCの音量					4ビット・データ			
R_{11}	エンベロープ周期	下位8ビット・データ							
R_{12}		上位8ビット・データ							
R_{13}	エンベロープ波形					CONT.	HOLD	ATT.	ALT.

（R_0〜R_5）音階は12ビットのデータで表現する．

（R_6）ノイズの平均周波数を指定する．

（R_7）各チャンネルから出す音源や，I/Oポートを指定する．

（R_8〜R_{10}）M=0のとき 下位4ビットが音量調節 M=1のとき エンベロープ作動

PSGのレジスタ

8.4.2　レジスタの説明

R_0 から R_5 は3個のオシレータの発振周波数を決定します．R_1 と R_0，R_3 と R_2，R_5 と R_4 はそれぞれ対で使用され，R_1，R_3，R_5 の下位4ビットと R_0，R_2，R_4 の下位8ビットの計12ビットで決定されます．発振周波数は次の式より算出できます．

$$f=\frac{f_{\mathrm{clock}}}{16\times D_1}$$

ここで FM-7 の $f_{\mathrm{clock}}=1.2288\mathrm{MHz}$ であり，D_1 はレジスタに書込む12ビットのデータです．

R_6 はノイズの平均周波数を決定するレジスタで，下位5ビット（D_2）で表します．

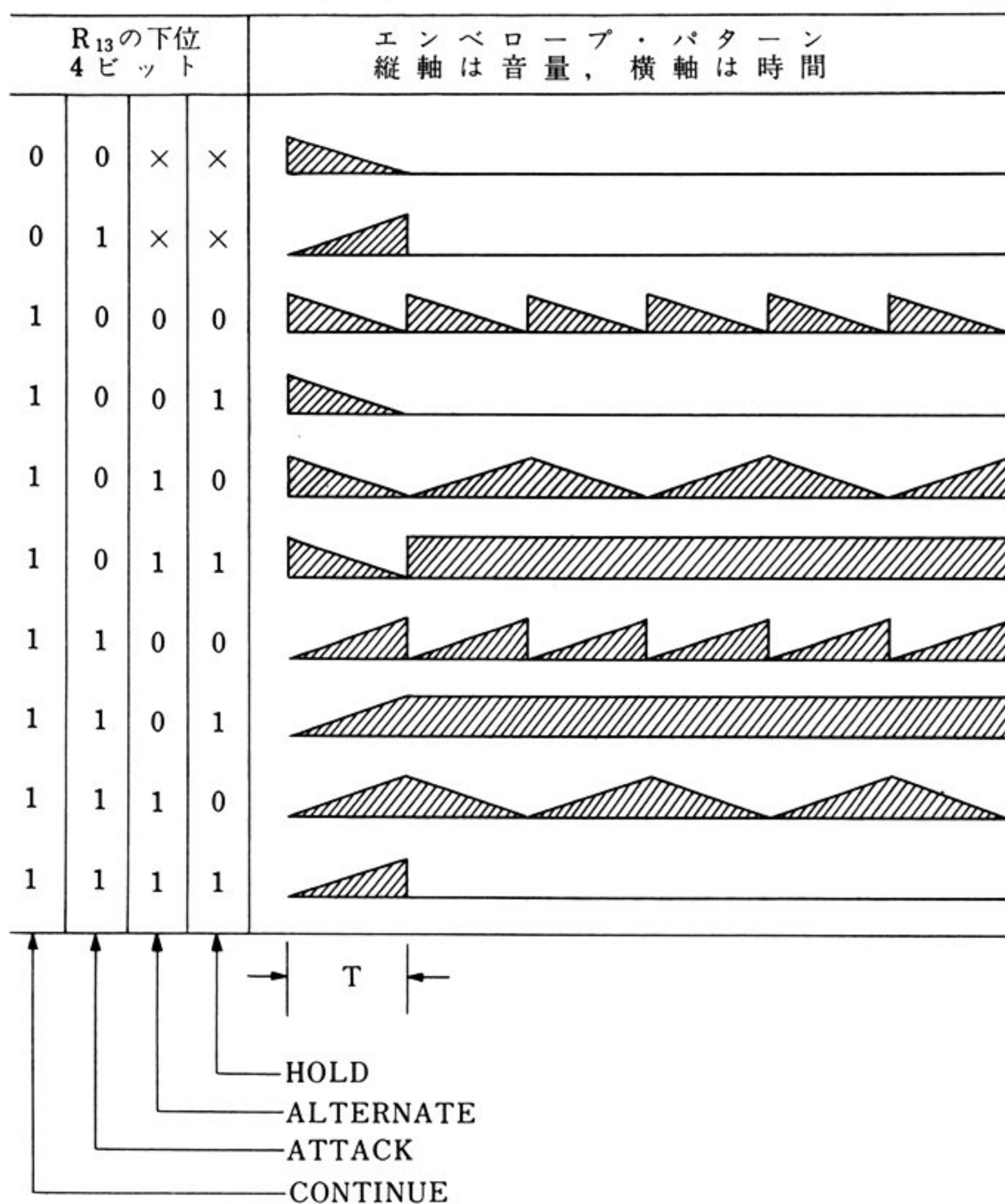
$$f=\frac{f_{\mathrm{clock}}}{16\times D_2}$$

R_7 は，各チャンネルのモード設定を行なうレジスタです．対応する各ビットを0にすることによりモードを設定できます．

R_8 から R_{10} は，各チャンネルの音量を設定する5ビット有効なレジスタです．音量は4ビットで指定し，最上位のビットはエンベロープモードとなり，音量の指定が無効となります．

R_{11} から R_{13} はエンベロープを制御するレジスタです．エンベロープの形式は R_{13} により指定し，それぞれのデータに対応したエンベロープを次の図に示します．

R13レジスタとエンベロープ・パターン

R13の下位4ビット				エンベロープ・パターン 縦軸は音量，横軸は時間
0	0	×	×	
0	1	×	×	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

エンベロープの長さTは R_{12} と R_{11} のレジスタのデータ D_3 で指定し，次式より計算できます．

$$T = \frac{256 \times D_3}{1.2288 \times 10^6} \quad (秒)$$

8.4.3 SOUND文の使用例

〈波の音〉

```
10 ' List 1 "Wave sound"
20 SOUND 7,&HF7 '    OSC. set
30 SOUND 6,&H05 '    Noise FRQ. set.
40 SOUND 8,&H10 '    Set Envelope. (part a)
50 SOUND 11,0 '      Envelope FRQ. set>
60 SOUND 12,55
70 SOUND 13,14'      Set enverope wave pattern.
```

〈飛行機の墜落音〉

```
10 ' List 2 "Explosion sound"
20 SOUND 7,&HFC  '      OSC. set
30 SOUND 8,12 '     Volume set of part A
40 SOUND 9,12 '         Volume set of part B
50 FOR I=1 TO 250 '     Fall sound
60 SOUND 0,I
70 SOUND 2,I
80 SOUND 1,0
90 SOUND 3,1
100 NEXT I
110 FOR I=1 TO 100
120 X=INT(RND(1)*256)  ' Crash sound
130 SOUND 0,X:SOUND 2,X
140 X=INT(RND(1)*2)+13:Y=X-INT(RND(1)*5)
150 SOUND 1,X:SOUND 3,Y
160 NEXT
170 SOUND 7,&HFF '      Sound stop.
180 END
```

〈タイプライタ音〉

```
10 ' List 3 "Typewriter"
20 SOUND 7,&HF7 '                  Noise set of part A
30 SOUND 8,16 '                    Set envelope.
40 SOUND 6,&H05 '                  Set noise FRQ.
50 SOUND 11,0 '                    Set envelope FRQ.
60 SOUND 12,8
70 A$=INKEY$:IF A$="" GOTO 70'   Input one character.
80 SOUND 13,0:PRINT A$; '          Output tone and a charactor.
90 GOTO 70
```

〈ジェット機のエンジン音〉

```
10 ' List 4 "Jet sound"
20 SOUND 7,&H35 '                      Part A:noise   Part B:Tone
30 SOUND 8,10:SOUND 9,10 '             Volume set.
40 SOUND 6,&H1F '                      Noise FRQ. set.
50 TNE=255 '                           Early tone FRQ.
60 TNE1=TNE MOD 256 '                  High ranking 1 byte set.
70 TNE2=TNE ¥ 256 '                    Low  ranking 1 byte set.
80 SOUND 2,TNE1:SOUND 3,TNE2 '         Set envelope FRQ.
90 SOUND 6,TNE/8.25 '                  Set noise FRQ.
100 A$=INKEY$:IF A$="U" OR A$="u" OR A$="D" OR A$="d" THEN 110 ELSE 100
110 IF (A$="D" OR A$="d") AND TNE<&H100 THEN TNE=TNE+1:GOTO 60
120 IF TNE<30 GOTO 100
130 TNE=TNE-1:GOTO 60
```

第9章　日本語表示

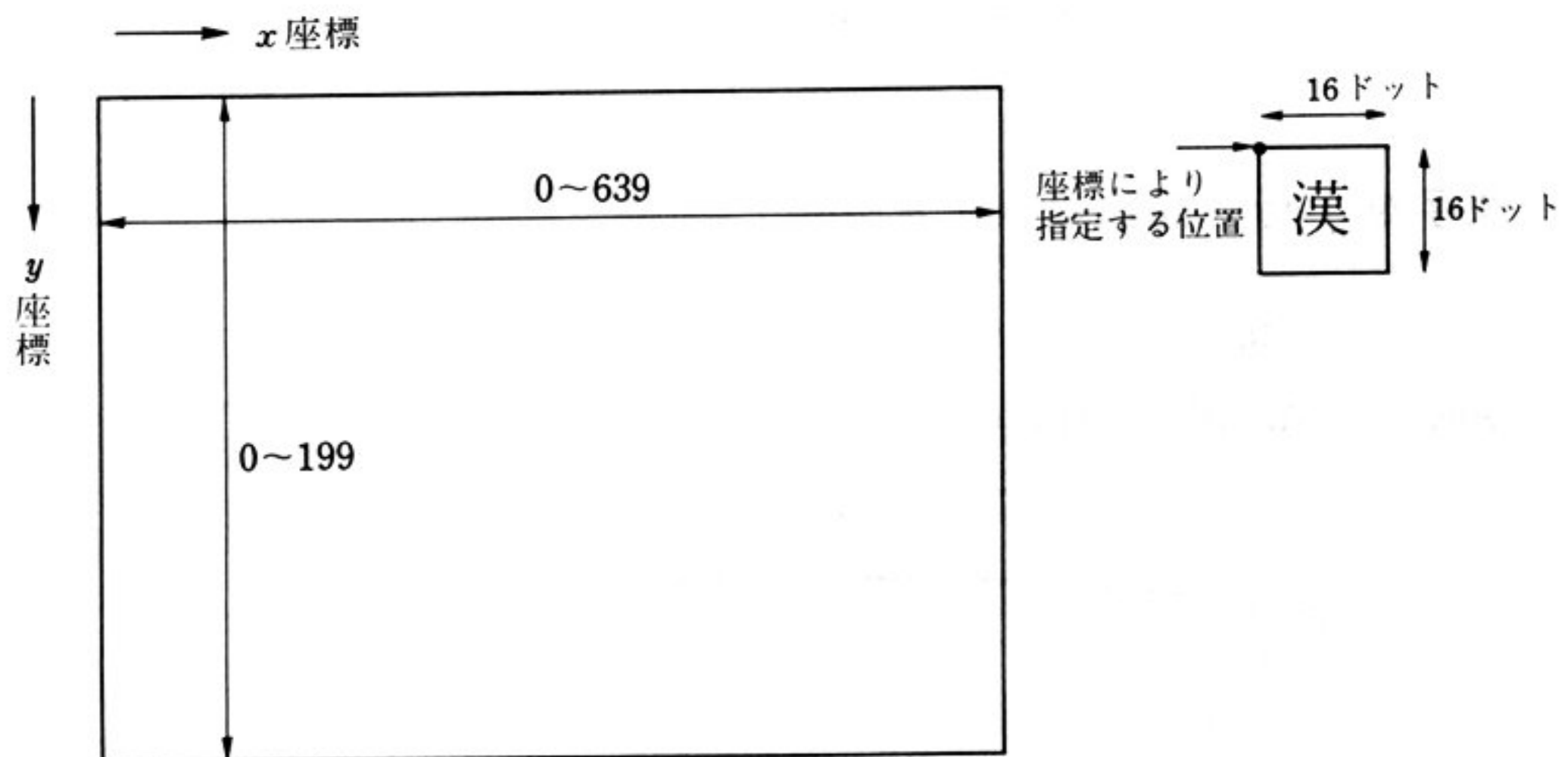
9.1　漢字の画面への表示方法

FM-7は，オプションの漢字ROMカードを，オプションスロットへ装着することにより，日本語の画面表示や，プリンタへの漢字出力ができます．

漢字キャラクタは，16×16ドットにより表示されますので，本体内のキャラクタの8×8ドットとは，大きさが異なり，しかも，グラフィック表示されますので，表示方法が異なります．

F-BASICにおいては，“PRINT @(プリントアットマーク)”命令を使用します．

以下に“PRINT @”についての説明を示します．

PRINT @ (x座標, y座標), 漢字コード

画面に対する座標

（1）　x座標，y座標

グラフィックにおける座標のことで，x方向へ0〜639，y方向へ0〜199の指定ができます．

注意することは，座標指定する場合，漢字キャラクタの左上端の座標を指定しますので，画面の端を指定すると，漢字キャラクタが，画面からはみ出てしまい，表示できない場合があります．

このため，座標はx方向へ0〜623，y方向へ0〜183の範囲で使用することになります．

(2) 漢字コード

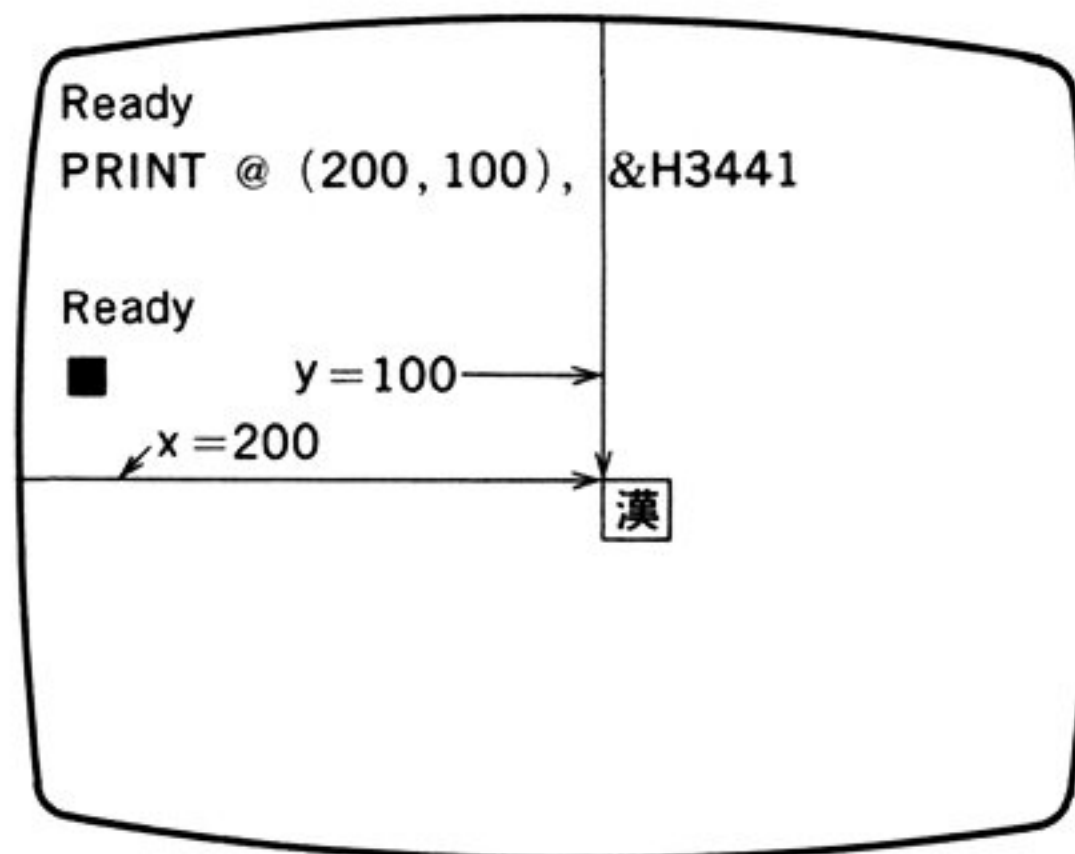
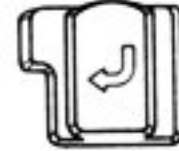
付録の漢字コード表より選択して，16進数または10進数により指定します．

漢字コード表は，すべて16進数により表示してあります．

9.2 漢字を1文字表示する場合

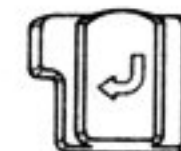
x方向200， y方向100の位置へ“漢”を表示します．

```
PRINT @ (200, 100), &H3441
```

```
Ready
PRINT @ (200,100), &H3441

Ready
■           y=100
    x=200
                        漢
```

9.3 漢字を1行へ複数表示する場合

x方向0，y方向150の位置より“漢字出力表示”と表示します．

```
PRINT @ (0, 150), &H3441, &H3B7A, &H3D50, &H4E4F,
                  &H493D, &H3C28
```

```
Ready
PRINT @ (0,150), &H3441, &H3B7A, &H3
D50, &H4E4F, &H493D, &H3C28

Ready
■

漢字出力表示
```

9.4 漢字を複数行 表示する場合

<プログラム例>

```
10 CLS
100 PRINT@(0,50),&H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
110 PRINT@(0,70),&H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
120 PRINT@(0,90),&H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
130 PRINT@(0,110),&H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
```

<画面表示>

```
Ready
■

漢字出力表示
漢字出力表示
漢字出力表示
漢字出力表示
```

上に示すプログラム例により，座標を変えることで，漢字を画面上の任意の場所へ表示できることが分ります．

9.5 漢字コードをデータとして扱う方法

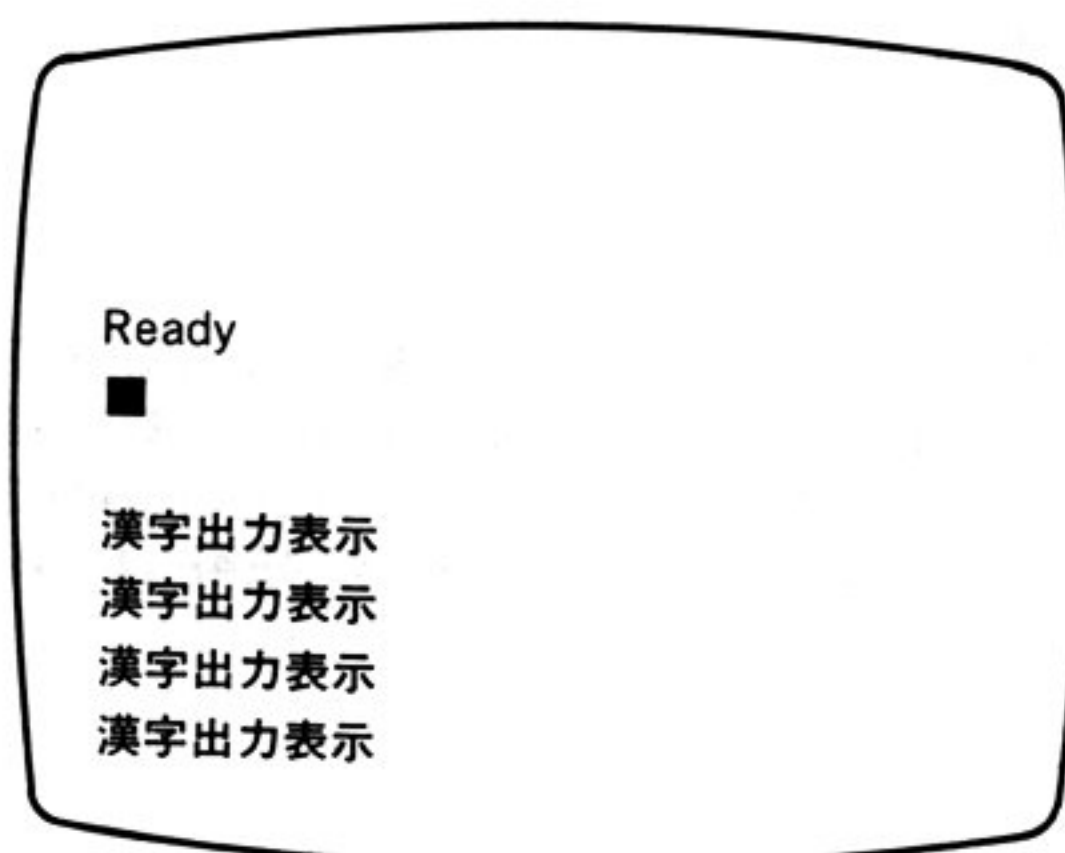
漢字を表示させる都度，PRINT 文を書くことは，プログラム作成上非常に時間もかかり，プログラムも大きくなってしまいます．

そこで，以下に漢字コードを DATA 文により，データとして扱う方法を示します．

<プログラム例>

```
10 CLS
20 FOR Y=50 TO 110 STEP 20
30     FOR X=0 TO 100 STEP 20
40         READ A%
50         PRINT@ (X,Y),A%
60     NEXT X
70 NEXT Y
80 END
100 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
110 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
120 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
130 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
```

<画面表示>

```
Ready
■

漢字出力表示
漢字出力表示
漢字出力表示
漢字出力表示
```

9.6 漢字をスクロールさせ表示する方法

FM-7は，画面へ出力するデータを持つRAM（ビデオRAM）へ，グラフィックデータとキャラクタデータの情報を格納しています．

つまり，キーボードより入力したキャラクタの情報と，漢字を表示するためのグラフィックの情報を同一RAM上において扱っているのです．

このため，キャラクタ画面のスクロールに伴ない，グラフィック画面も同様にスクロールを行ないます．

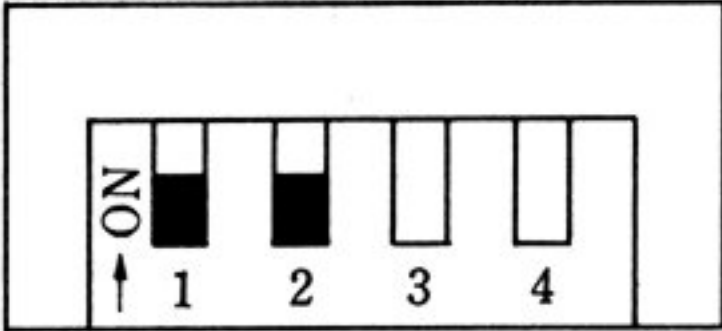
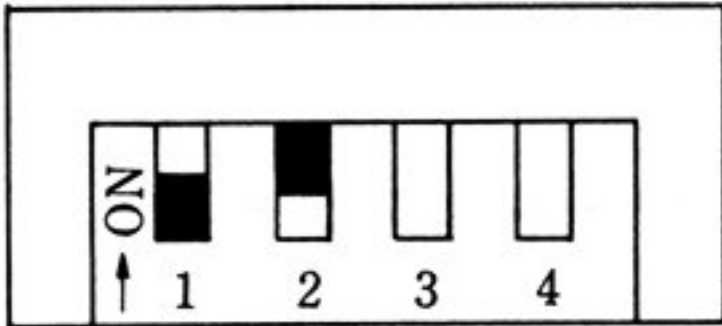
この機能を利用し，漢字をスクロールさせながら画面へ表示する例を示します．

<プログラム例>

```
10 CLS:LOCATE 0,20
20 FOR I=1 TO 4
30    FOR X=0 TO 100 STEP 20
40       READ A%
50       PRINT@ (X,180),A%
60    NEXT X
70   PRINT:PRINT
80 NEXT I
90 END
100 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
110 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
120 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
130 DATA &H3441,&H3B7A,&H3D50,&H4E4F,&H493D,&H3C28
```

＜画面表示＞

```
漢字出力表示
漢字出力表示
漢字出力表示
漢字出力表示
Ready
■
```

第10章　その他の機能

10.1　ブートローダ機能

通常，コンピュータシステムの起動は，IPL（Initial Program Loader：イニシャルプログラムローダ）の実行により，行なうことができます．

IPLは，システムの制御を行なうプログラム（F-BASICや各種DOS）のメモリへのロードや，制御を渡すためのプログラムであり，また，IPLを，メモリへロードし，実行させるためのプログラムを〝ブートプログラム〞といい，このプログラムの機能を〝ブートローダ機能〞といいます．

FM-7は，本体内にブートローダ機能を有するROMが実装されており，電源投入後，あるいはシステムリセット後，このROM内のブートプログラムが働き，ブートローディングを行ないます．

さらに，ブートローダ機能には，2種類のモードがあり，使用するOS（operating system：オペレーティングシステム）により，モードの設定を行ないます．モードの設定は，本体背面のディップスイッチにより行ないます．

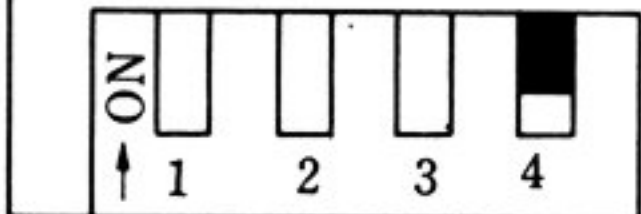
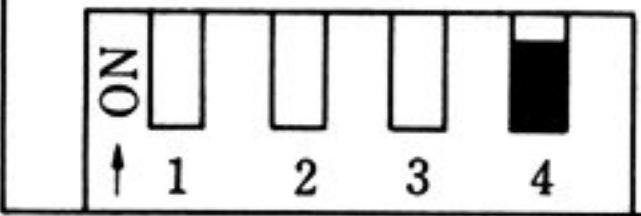
モード	機能	ディップスイッチ
ROMモード DISKモード	F-BASIC V3.0 ROMモードまたはDISKモードを使用する場合，設定します．	1，2番共ONにします． ON↑ 1 2 3 4
DOSモード	FM-7用のDOSを使用する場合，設定します．	1番ON，2番OFFにします． ON↑ 1 2 3 4

出荷時は，ROMモードとなっています．

ROMモードにおいて，ブートローダは必ずミニフロッピィディスクユニットが接続されていることを確認します．ミニフロッピィディスクユニットが接続され，さらに，ドライブ番号0番へシステムディスク（F-BASIC V3.0）がセットされることにより，IPLを，システムディスクよりロードし，実行するとDISKモードとなります．

ミニフロッピィディスクが接続されていなければ，ROMモードに設定されます．

10.2 動作クロック周波数の設定

FM-7は，動作クロック周波数を変更することにより，実行速度を変更できます．

実行速度の変更は，本体背面のディップスイッチにより，標準動作モードおよび，倍速動作モードに設定します．

標準動作モードは，FM-8と互換性を持ち，FM-8において作成されたプログラムや，FM-8用に販売されている各種ソフトウェアを利用する際，設定して下さい．

倍速動作モードは，プログラムの開発を行なう場合に，処理速度の高速化がはかれます．

なお，この動作モードにおいて，FM-8用プログラムを実行することもできますが，操作上で不具合を生じた場合は，一度，システムの電源をOFF状態とした後，ディップスイッチを標準動作モードに変更して下さい．

以下にディップスイッチの設定を示します．

動作クロック周波数	ディップスイッチ
倍速動作モード （8 MHz）	4番OFFにします． ON ↑ 1 2 3 4
標準動作モード （4.9MHz）	4番ONにします． ON ↑ 1 2 3 4

このディップスイッチは，1，2番がブートローダ，3番は予備として使用されておらず，4番が動作クロック周波数の設定用となっております．

出荷時は，倍速動作モードとなっております．

ディップスイッチは，常にブートローダと動作クロック周波数との組合わせにより設定されます．

10.3 ウォームスタート

電源や，リセットスイッチの投入により，システムのリセットが行なわれ，それまで存在したメモリ上のプログラムやデータはすべてクリアされます．しかし，ROM／DISKモード使用時において，メモリ上のプログラムやデータ等をクリアせずに，メモリ上へ残したまま，本体および周辺装置のリセットを行なう方法を，"ウォームスタート（warm start）"といいます．

ウォームスタートの方法は，Breakキーを押しながら，リセットスイッチを押下し，その後，Breakキーを離します．すると，画面がクリアされ，周辺装置のリセットが行なわれ，画面へ"Abort"と

"Ready" を表示した後，F-BASIC のコマンドレベルにもどります．以上がウォームスタートです．

10.4 プログラム作成上の注意

・F-BASIC V1.0 のシステムにおいて，F-BASIC V3.0 の命令中で，V1.0 に存在しない命令を用いてプログラムを作成し，バイナリ形式によりセーブした場合，その後，V3.0 のシステムのもとで実行させても，その命令により，エラー（syntax error 等）が発生します．

この原因は，V1.0 のもとでプログラムを作成する際，入力された BASIC ステートメントは，中間コードに変換されますが，コードが存在しないため，V1.0 では，命令として認識せずに、別のコードとなってしまうためです．

たとえば，V3.0 における "LPRINT" 命令を，V1.0 のシステムのもとで使用した場合，V1.0 は，"LPRINT" という変数名として認識します．そして，バイナリ形式によりセーブし，V3.0 のシステムのもとで実行しても，変数名として実行しますので，"LPRINT 命令" としての機能は働かないのです．

これを防ぐためには，アスキー形式により，セーブを行なって下さい．F-BASIC V1.0 と F-BASIC V3.0 の共通の命令は，バイナリ形式の受け渡しであっても実行できます．

・F-BASIC V3.0 において，省略形を使用できる命令と，できない命令がありますので注意して下さい．

<例>

LPRINT の場合

PRINT 命令は，"?" により代用できますが，これを LPRINT 命令において使用しますとエラー（syntax error）が発生します．

プログラム作成上，"LPRINT" を "L?" と入力し，LIST 命令により画面へリストを表示すると，"LPRINT" と表示しますが，この命令を実行するとエラーとなります．

```
Ready
10 L? "AAA"
LIST

10 LPRINT "AAA"

Ready
RUN

Syntax Error In 10
Ready
```

この場合は，"L" という変数名と，"PRINT" という命令に分かれ，"LPRINT" とはならないためです．

第11章 ファイル

11.1 ファイルの概念

F-BASICにおいて，プログラムやデータを補助記憶媒体（カセットテープやフロッピィディスク）へ記録したり，その記録したものを読むという処理や，入出力装置とのデータの入力，出力は，全て〝ファイル〟という概念を用いています．

この〝ファイル〟というのは，私達が日常，書類等をとじるために使用しているファイルのことです．

私達は，関連性のある書類等をまとめて，ファイルへとじ，ファイルの表紙や背表紙に表題をつけて書類棚へ保管します．そして，必要に応じて表題により，目的のファイルを探して，その中の書類を使用します．

これを，F-BASICにおいて考えると，プログラムやデータが書類となり，プログラムやデータをファイルへ，表題であるファイルの名前（ファイル名）をつけてとじ，書類棚にあたる補助記憶媒体へ保管し，プログラムやデータを使用する場合もファイル名により探し，使用するということになります．

入出力装置に対するデータの入力，出力においては，ある人が書類をファイルへとじると，その書類を他の人が持ち出し，他の処理をするということになります．つまり，入力装置よりデータをファイルへとじると，出力装置がそのデータをファイルから持ち出し，データの出力を行なうと考えることができます．

以上のように，F-BASICでは，プログラムやデータの保管や入出力において，常にファイルを介して行なっているということがいえるのです．

補助記憶媒体に対するファイルの概念

補助記憶媒体

プログラム

データ

保管

ファイル

とじる

プログラム，データ

データ

入出力装置に対するファイルの概念

出力装置

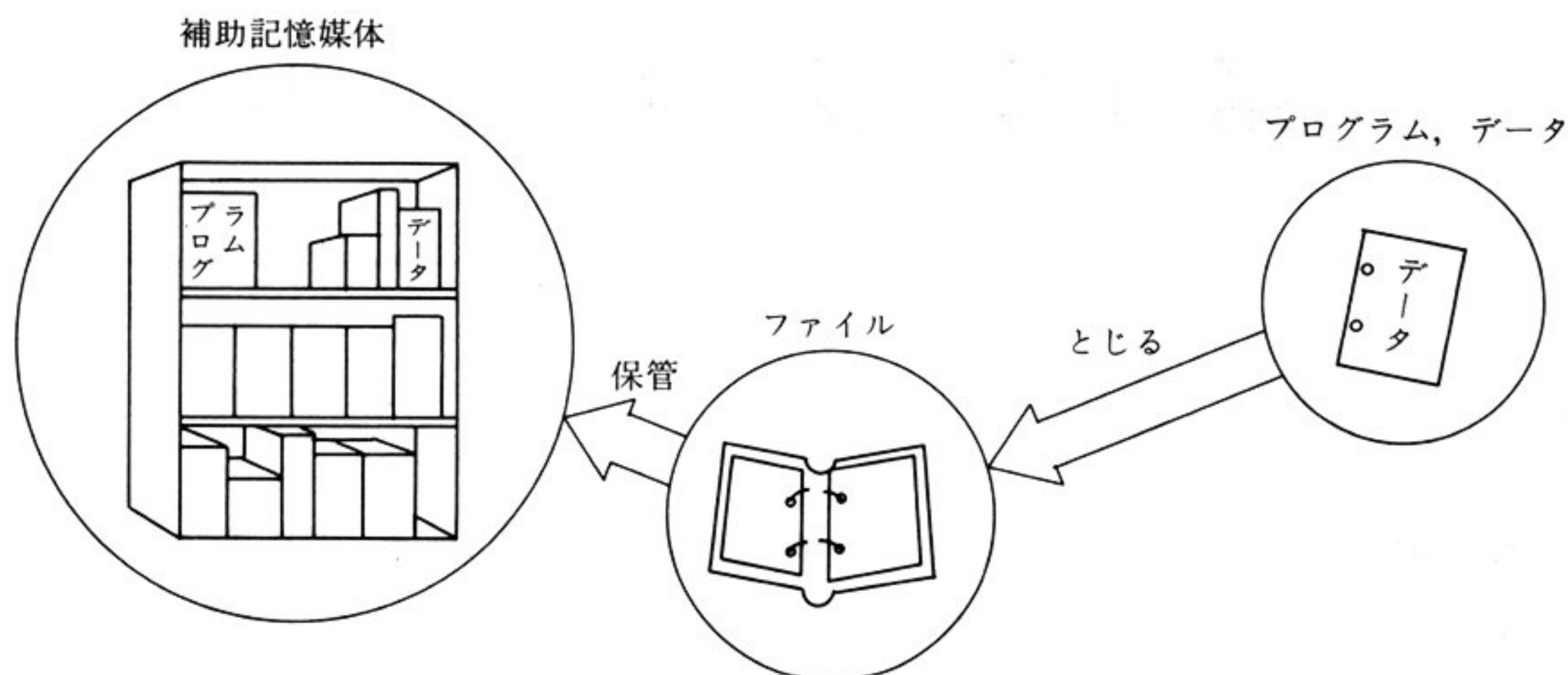
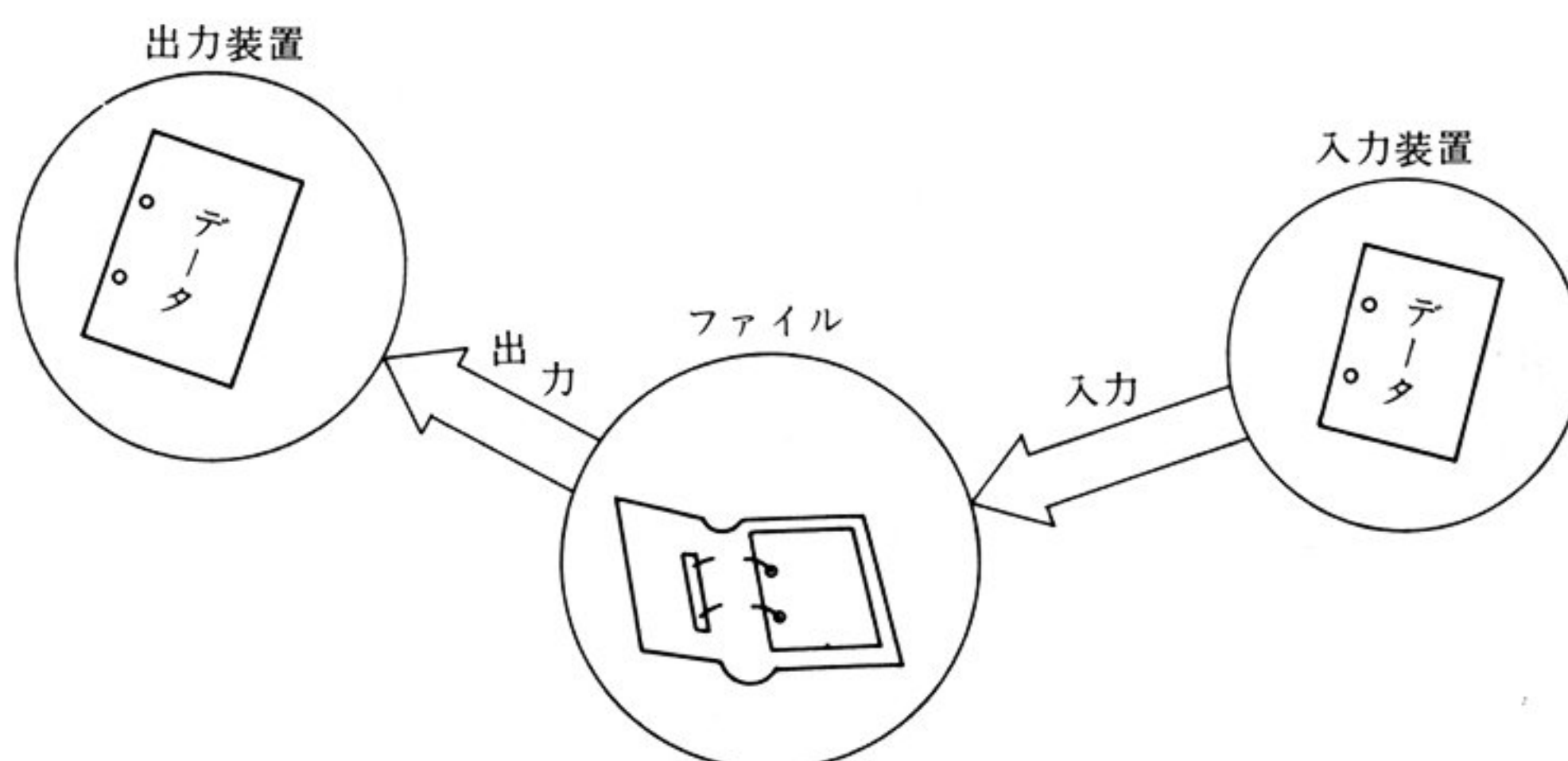
データ

出力

ファイル

入力

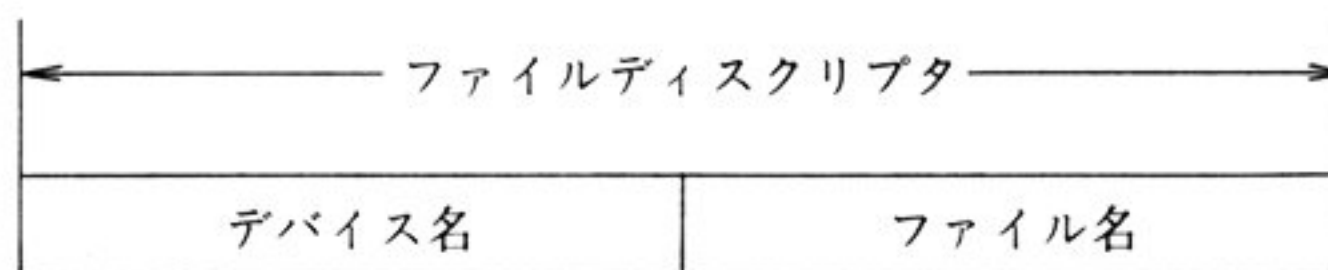
入力装置

データ

11.2 ファイルディスクリプタ (file descriptor)

補助記憶媒体において，プログラムやデータを使用するために〝ファイルディスクリプタ〟を指定します．〝ファイルディスクリプタ〟は，〝ファイル名〟と〝デバイス名〟により構成されます．

〝ファイル名〟は，プログラムやデータの名前ですが，〝デバイス名〟は，補助記憶装置（媒体）に対する名前です．つまり，〝ファイルディスクリプタ〟により，どういう書類棚のどのファイルを使用するという指定を行なうのです．

ファイルディスクリプタは，必ずデバイス名を先に，ファイル名を後に記述するという規則があります．

ファイルディスクリプタ	
デバイス名	ファイル名

11.2.1 ファイル名 (file name)

ファイル名は，8文字までの英数字，カナ，記号により任意に指定できます．後でどのようなプログラムやデータなのか，わかりやすい名前にしておくと便利です．たとえば，10月10日に作成したデータに対し，〝DATA 1010〟としておけば，後で，「このデータは10月10日に作られたものだな.」とわかります．

しかし，ファイル名において使用が禁止されている記号がありますので注意して下さい．以下に使用禁止の記号を示します．

記号	名称
"	ダブルクォーテーション (double quotation)
(	左かっこ (left parenthesis)
)	右かっこ (right parenthesis)
:	コロン (colon)
;	セミコロン (semicolon)
,	コンマ (comma)

以上の記号は，F-BASICの文法において使用されますので，ファイル名として使用することはできません．

11.2.2 デバイス名 (device name)

F-BASICにおいては，補助記憶装置および入出力装置の指定は，〝デバイス名〟により行ないます．〝デバイス〟とは，補助記憶装置および，入出力装置を指し，〝デバイス名〟は，各装置に対して決められている記号です．

以下にFM-7において使用可能な装置と，その装置に対するデバイス名を示します．

装置名称	入力	出力	デバイス名	デバイス名の扱い
オーディオカセット	○	○	CAS 0：	ROM モードにおいては省略できます．
ミニ フロッピィディスク	○	○	0：	DISK モードにおいては省略できます．
			1：	2ドライブのミニフロッピィディスクユニットにおいて扱うことができます．
プリンタ	×	○	LPT 0：	機種に対しては全て共通です．
回線制御 インタフェース	○	○	COM 0：	オプションの RS-232C インタフェースカードが必要です．
			COM 1： COM 2： COM 3： COM 4：	FM-7 と I/O 拡張ユニットまたは，システム拡張ユニットを使用して RS-232C インタフェースモジュールを接続した場合に使用できます．
キーボード	○	×	KYBD ：	キーボードよりデータ入力を示します．
スクリーン(画面)	×	○	SCRN ：	画面への出力を示します．

11.3 ファイルの扱い(ファイルの種類)

F-BASIC において，プログラムとデータに対し，ファイルの概念は共通ですが，ファイルに対する扱い方は異なります．

プログラムは，1つのプログラムが1つのファイルと対応しますが，データは，1つのファイルと多くのデータが対応しますので，プログラムおよびデータのファイルに対する入出力方法が異なってくるのです．

プログラムにより作成されたファイルをプログラムファイルと呼び，データにより作成されたファイルをデータファイルと呼びます．これらのファイルについても種類があり，扱い方も異なってきます．

以下にファイルの種類を示します．

ファイルの名称		特　　徴
プログラムファイル	BASIC プログラム ファイル	F-BASIC プログラムにより作成されたファイルです．
	機械語プログラム ファイル	機械語プログラムにより作成されたファイルです．
データファイル	シーケンシャル ファイル	データの扱いを連続的に行なうファイルです．
	ランダム ファイル	データを記録している位置が管理されているファイルです．

入出力装置に対するファイルの扱いは，データ ファイルと同様です．データの入出力は連続的に行なわれますので，シーケンシャルファイルと同じ扱いになります．データを出力する場合，出力先が補助記憶媒体であれば，データが記録され，シーケンシャル ファイルとなります．

11.3.1 プログラム ファイル(ロード・セーブ)

プログラムは，キーボードより入力し，本体内のメモリ上に作成します．メモリ上に作成したプログラムを補助記憶媒体へ記録し，プログラム ファイルとすることを〝セーブ（SAVE）する〟といい，記録されたプログラムをメモリ上へ展開し，実行可能にすることを〝ロード（LOAD）する〟といいます．

F-BASIC におけるロード・セーブに関する命令の記述方法を以下に示します．

条件			記述形式
ロードする場合	BASIC プログラム ファイル		LOAD〝ファイルディスクリプタ〟
	機械語プログラム ファイル		LOADM〝ファイルディスクリプタ〟[，オフセット値]
セーブする場合	BASIC プログラム ファイル	バイナリ形式	SAVE〝ファイルディスクリプタ〟
		アスキー形式	SAVE〝ファイルディスクリプタ〟，A
	機械語プログラム ファイル		SAVEM〝ファイルディスクリプタ〟，開始番地，終了番地，入口番地

F-BASIC により作成したプログラムをセーブし BASIC プログラム ファイルとする場合，アスキー形式とバイナリ形式とが指定できます．

アスキー形式とは，プログラムの一文字ずつを，アスキーコードによりセーブし，プログラムファイルとする方法です．アスキーコードによりプログラムファイルが，作成されていることより，データファイルとして利用できるという利点があります．

アスキー形式に対し，バイナリ形式とは，プログラムを，アスキーコードではなく，〝中間コード〟という，BASIC ステートメントを簡略化したコードにより，セーブを行なう方法です．

中間コードとは，プログラムがメモリ上に展開された形であり，実行や処理も，このコードにより行なわれます．

つまり，バイナリ形式により，プログラムをセーブするということは，メモリ上に展開されている形のまま，プログラムを記録することになります．

また，長いプログラムにおいて，バイナリ形式のプログラムファイルは，アスキー形式のものと比べ，補助記憶媒体中を占める領域が少なくてすむという利点があります．

しかし，F-BASIC において，プログラム ファイルがアスキー形式でないと命令が実行できない場

合がありますので注意して下さい.

・〝MERGE（マージ）〞命令を実行する場合，マージされるプログラムはアスキー形式により，セーブされていなければなりません.

・〝CHAIN（チェイン）〞命令において，MERGE指定をする際，メモリ上のプログラムにおける最大の行番号より，チェインされるプログラムの先頭の行番号が小さい場合，チェインされるプログラムは，アスキー形式によりセーブされていなければなりません.

11.3.2 データ ファイル

（1） オープン・クローズ

データ ファイルは，プログラム ファイルと違い，多くのデータを1つのファイルにより入出力が行なえますので，ファイルに対する扱い方がプログラム ファイルとは異なります.

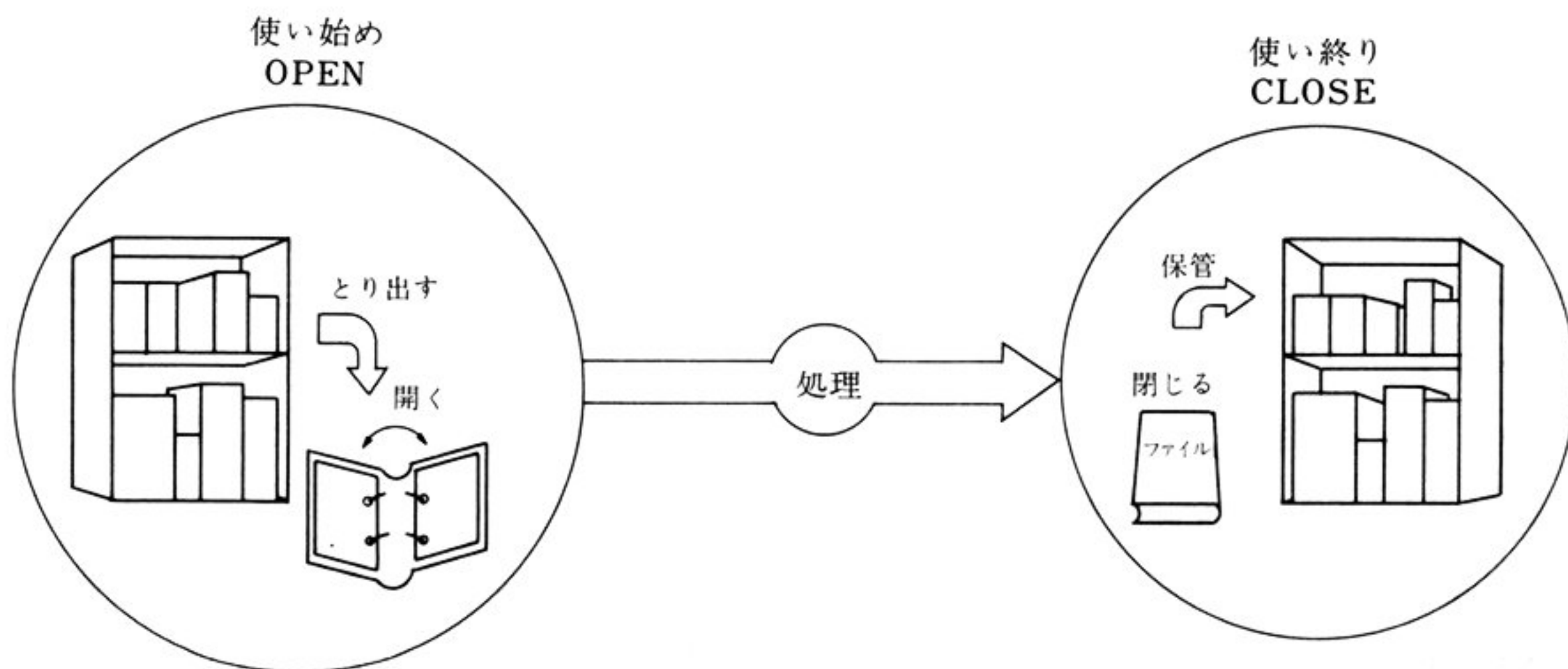
つまり，プログラム ファイルにおいて，プログラムは一個のデータとして扱いますので，データの始めと終りをF-BASICが認識でき，それによりファイルを開いたり，とじたり保管したりといった処理を，F-BASICが行ないます.

ところが，データ ファイルにおいては，多くのデータを扱い，データの数は使用者により決定されますので，ファイルの使い始めや，使い終りを指定しなければなりません. 使いはじめが〝OPEN（オープン）〞命令であり，使い終りが〝CLOSE（クローズ）〞命令です.

つまり，OPEN命令によりファイルを指定し，開いて，書類をとじることができるよう準備し，CLOSE命令により，とじ終ったファイルを書類棚へ返却し，処理の終了を示します.

OPEN命令を使用し，ファイルを開き処理をし終れば，必ずCLOSE命令によりファイルを閉じなければなりません. ファイルが開いたままだとそのファイルをこわすことにもなりかねませんのでOPEN命令を使用した後は，必ずCLOSE命令によりファイルを閉じるよう注意して下さい.

使い始め
OPEN
とり出す
開く
処理
使い終り
CLOSE
保管
閉じる
ファイル

OPEN命令は，ファイルを使用するための準備を行ないますが，この命令において，ファイルに対する処理や，ファイルの名前を指定します．

以下にOPEN命令の記述形式を示します．

OPEN "モード", #ファイル番号, "ファイルディスクリプタ"
または，デバイス名

"モード"とは，ファイルをどのように使用するかの指定を行ないます．以下に"モード"の種類とその処理内容を示します．

モード	ファイルの指定	処理内容
I (Input：インプット)	ファイルディスクリプタ	シーケンシャル ファイルより，データの入力を指定します．
	デバイス名	デバイスより，データの入力を指定します．
O (Output：アウトプット)	ファイルディスクリプタ	新しくシーケンシャル ファイルを作成し，そのファイルへデータの出力を指定します．
	デバイス名	デバイスへ、データの出力を指定します．
A (Append：アペンド)	ファイルディスクリプタ	ミニフロッピィディスクにおけるシーケンシャル ファイルに対し、データの追加を指定します．
R (Random：ランダム)	ファイルディスクリプタ	ミニフロッピィディスクにおいて，ランダム ファイルに対する処理を指定します．

(入力・出力は，すべてFM-7本体に対するものです)

OPEN文において，ファイルディスクリプタまたは，デバイス名により，使用するファイルの指定を行ないます．ファイルディスクリプタであれば，補助記憶媒体におけるファイルであり，デバイス名であれば，入出力装置に対するファイルであることを示しますが，さらに，ファイルの扱いを簡単にするため，ファイルに番号を付けます．この番号を"ファイル番号"といい，ファイルを扱う際には必ず，ファイル番号の指定を行ないます．ファイル番号に付ける"#(ナンバー記号：number sign)"は省略してもかまいません．

1つのデバイスに対し，複数のファイルを使用する際，そのファイルは，ファイル番号により区別できるのです．よって，同じファイル番号を持つファイルを同時に開くことはできません．また，ファイル番号は1～16まで指定できます．

以下に，デバイスに対し同時に開けるファイルの数を示します．

装置名称	デバイス名	同時に開ける数	注　意　事　項
カセットテープ	CAS0：	1	シーケンシャル ファイルに対する入力または，出力の際に使用します．
ミニフロッピィディスク	0： 1：	ファイルの合計が15まで許されます．	F-BASICのDISKモード起動時の初期設定により，数が制限されます．
プリンタ	LPT0：	1	出力のみです．
RS-232C	COM0： COM1： COM2： COM3： COM4：	それぞれのポートに対し2つのファイルが開けます．	各ポートにおいて，入力と出力の両方に対するファイルを同時に使用できます．
キーボード	KYBD：	1	入力のみです．
スクリーン(画面)	SCRN：	1	出力のみです．

上の表において，ミニフロッピィディスクの注意事項に記されているように，ミニフロッピィディスクにおいて使用できるファイルの数は，DISKモード起動時の初期設定により制限されます．

初期設定時の画面は以下のように出力されます．

```
DISK VERSION
How many disk drives          ?
How many disk files (0-15) ?
```

ここで，How many disk files (0−15) ?というメッセージは，「ミニフロッピィディスクに対するファイルを，同時に，合計いくつ使用するのですか？」ということです．この設定によって，入力された数より多く，同時にミニフロッピィディスクに関するファイルは使用できません．

初期設定により，ミニフロッピィディスクのファイルに対する〝バッファ〟が，入力された数確保されます．

〝バッファ〟とは，ファイルに対するデータの入出力を行なう際，データを一時格納しておき，そのデータを効率よく利用するために設けられた領域です．データは常にこのバッファを通り入出力されます．

ミニフロッピィディスクを扱う上で，OPEN命令におけるファイル番号の指定とは，ファイルを指定すると共に，そのファイルに対するバッファを割当てることです．

このとき，バッファは，メモリ上に，271バイトずつ確保されますので，その分，使用可能なメモリ領域が減少します．

よって，初期画面において，ファイル数の設定は，必要最小限とした方が，ユーザ領域を大きく使用できます．

ファイルを3つ使用する場合，それぞれのファイルにバッファを割当てると，バッファは3つ必要です．しかし，この3つのファイルを同時に使用する必要がなければ，1つのバッファを設定して，3つのファイルで順番に使用すると，バッファは1つしか必要ないのです．

これは，1つのファイルに対するOPEN命令でバッファを指定したそのファイルに対する処理を終了しCLOSE命令を実行すると，今まで使用していたバッファは解放されますので，次のファイルに対するOPEN命令により新しくバッファを割当てるというように処理を行ないます．

バッファを1つずつ割当てた場合（バッファは3つ必要です）．

データ —OPEN→ バッファ → ファイル
データ —OPEN→ バッファ → ファイル
データ —OPEN→ バッファ → ファイル
CLOSE

1つのバッファを使用した場合（バッファは1つですみます）．

データ OPEN CLOSE
データ OPEN CLOSE → バッファ → ファイル / ファイル / ファイル
データ OPEN CLOSE

（2） データ ファイルの種類

データ ファイルにおいて，データを連続的に記録して作成したファイルを，〝シーケンシャル ファイル〟といい，シーケンシャル ファイルにおいては，記録した順番にデータを読みます．また，データに〝レコード番号〟という番号を付け，個々のデータをレコード番号により扱うことのできるファイルを，〝ランダム ファイル〟といいます．

つまり，1枚の書類へすべてのデータを並べて記録し，ファイルにとじるのがシーケンシャル ファイルです．

これに対し，1枚の書類へ記録するデータを制限し，書類にページを付けて何ページもファイルに

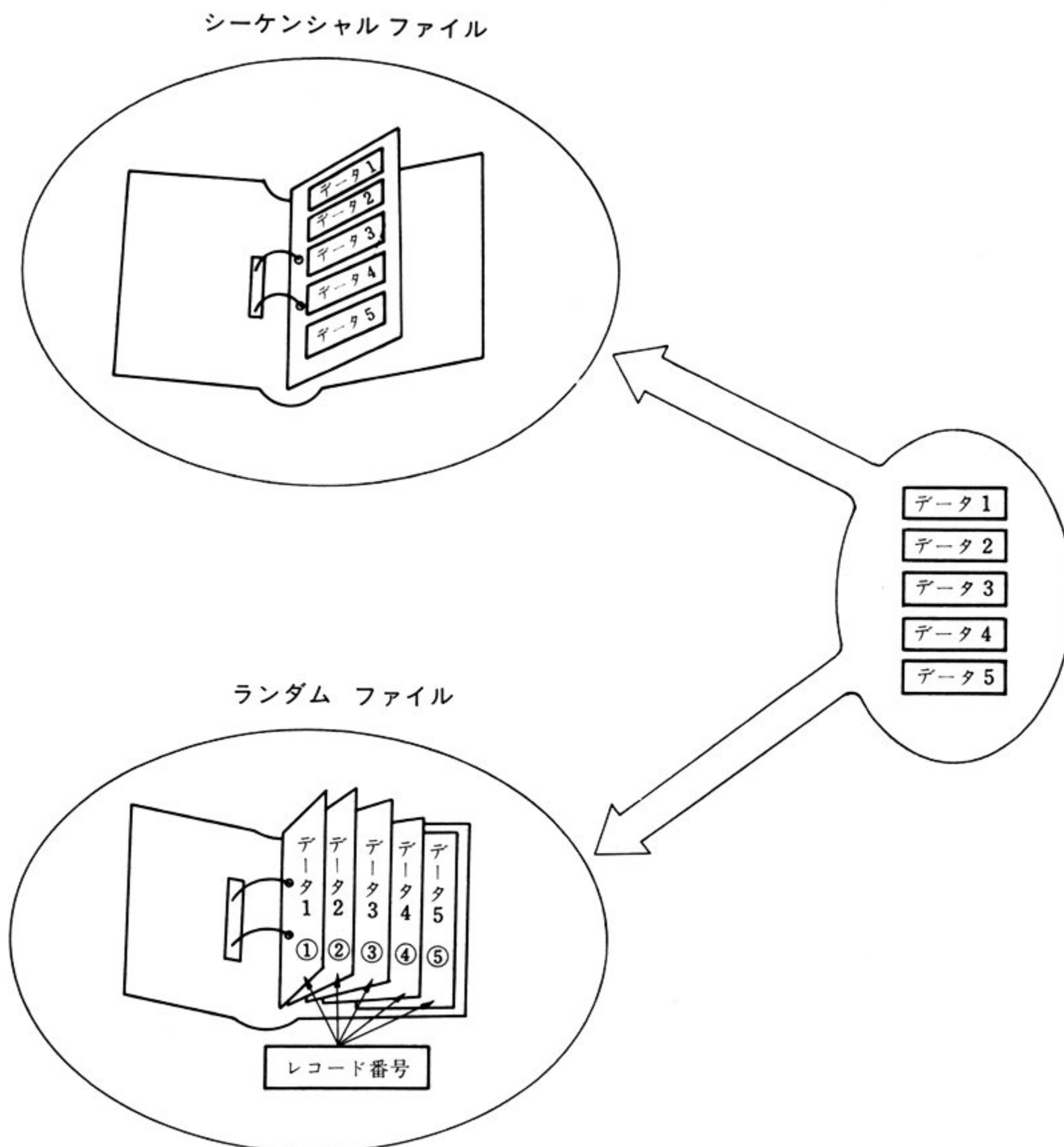
とじるのがランダム ファイルであり，このページを指定することにより，ページごとにデータを扱うことができます．このページが〝レコード番号〟になります．

シーケンシャル ファイル

データ1
データ2
データ3
データ4
データ5

ランダム ファイル

データ1 ①
データ2 ②
データ3 ③
データ4 ④
データ5 ⑤

レコード番号

それぞれのファイルは，以下に示す補助記憶媒体により，扱うことができます．

ファイルの種類	補助記憶媒体
シーケンシャル ファイル	カセット テープ ミニフロッピィ ディスク
ランダム ファイル	ミニフロッピィ ディスク

他の入出力装置に対する扱いは，シーケンシャル ファイルと同じです．プログラムを使用した説明は，各デバイスについての説明を行なっている章を参照して下さい．

第12章　カセットテープ

12.1　テープレコーダの扱い方

FMデータレコーダや，市販のテープレコーダの本体との接続は，本体付属のケーブルを使用します.

このケーブルのレコーダ側には，3つの端子があります.

・赤色の端子

出力用の端子です．マイクロホン用（MICまたはAUX）端子へ接続します.

・白色の端子

入力用の端子です．イヤホン用（EAR）端子へ接続します.

・黒色の端子

リモートコントロール用の端子です．リモート用（REM）端子へ接続します.

この端子を接続することにより，F-BASICの命令でカセットテープのスタート，ストップを制御できます.

プログラムやデータを保存するためのカセットテープは，市販されているものを使用します.

テ- プレコーダは，市販されているものでかまいませんが，FM-7との相性の問題が生じます.

これは，テープレコーダから出力される波形と，FM-7で使用している波形との違いからくるものです.

この点を考えて，FMデータレコーダ（MB 27501）は設計されておりますので，プログラムのロード，セーブ，データの入出力の際，エラーの発生が少なくなります.

また，ボーレート（baud rate）は，1600ボーです．CPUに対するクロックの設定を変更した場合も，F-BASICが認識し，常に1600ボーに設定します.

ボーレートとは，データを転送する速さの単位でもあり，1600ボーとは，1秒間に1600ビット転送することを示します.

12.1.1　家庭用テープレコーダに対する調整法

家庭用テープレコーダを使用する際の調整方法を以下に示します.

(1)　本体とテープレコーダを接続して，本体付属のカセットテープをA面が見えるように，テープレ

コーダへ入れて下さい.

(2) テープレコーダ，ディスプレイの電源を投入します.

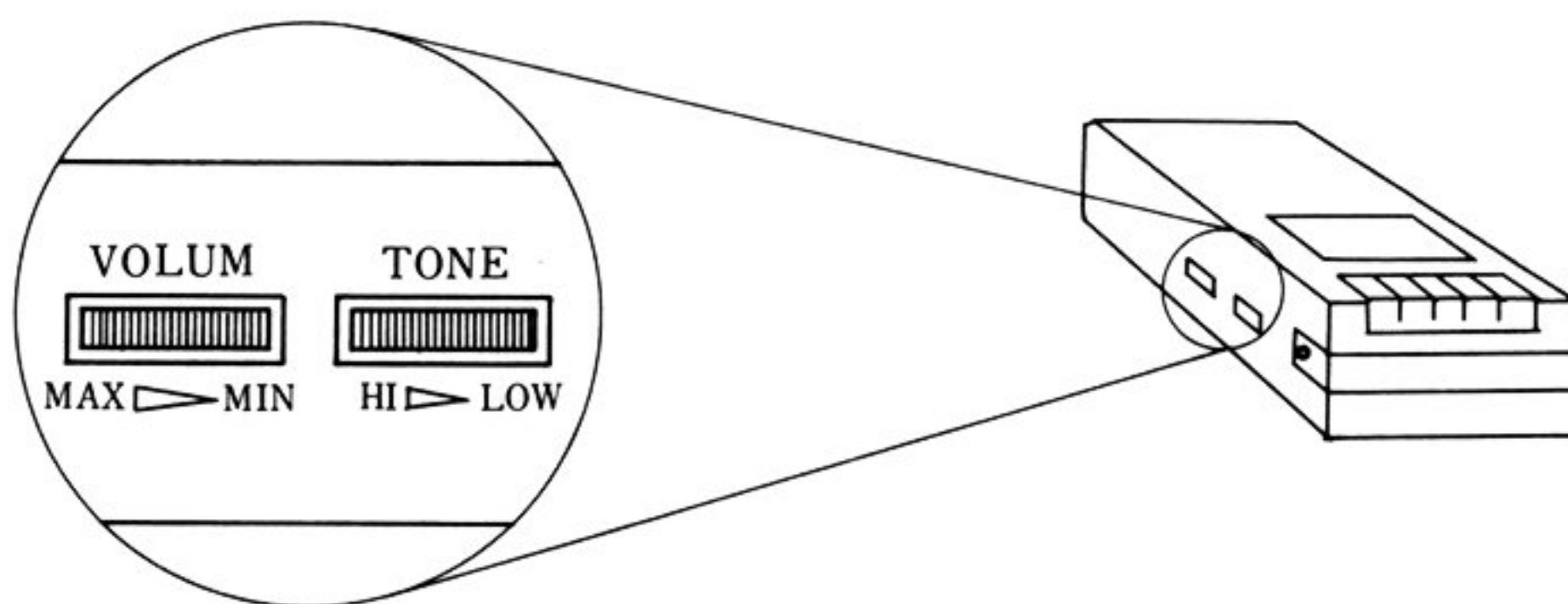
(3) テープレコーダのリモート端子をはずして，カセットテープを巻き戻しておきます.

(4) 本体の電源を投入して，F-BASIC V3.0 ROM モードを起動して下さい.

起動時の画面表示

```
FUJITSU F-BASIC Version 3.0
Copyright (C) 1981 By FUJITSU/MICROSOFT
nnnnn Bytes Free

Ready
█
```

(5) テープレコーダの音量（ボリューム）を最大にし，音質（トーン）を中間の位置に合わせます.

VOLUM　TONE
MAX ▷ MIN　HI ▷ LOW

(6) テープのプログラムをロードします. 以下のように入力して下さい.

LOAD ↵

(7) 画面にSearchingのメッセージが出力されますので，PLAY ボタンを押し，テープを走行させます.

(8) Found DEMO1 とメッセージを出力した後, Ready と表示し，コマンド受付け状態に戻れば，プログラムのロードを終了したことを示します.

(9) PLAY ボタンを押した後，Device I/O Error とエラー表示する場合がありますので, 以下のように調整して下さい.

(10) テープを巻き戻し，止めておきます.

(11) 音量をやや下げ，再度プログラムのロード（6の操作）を行ないます.

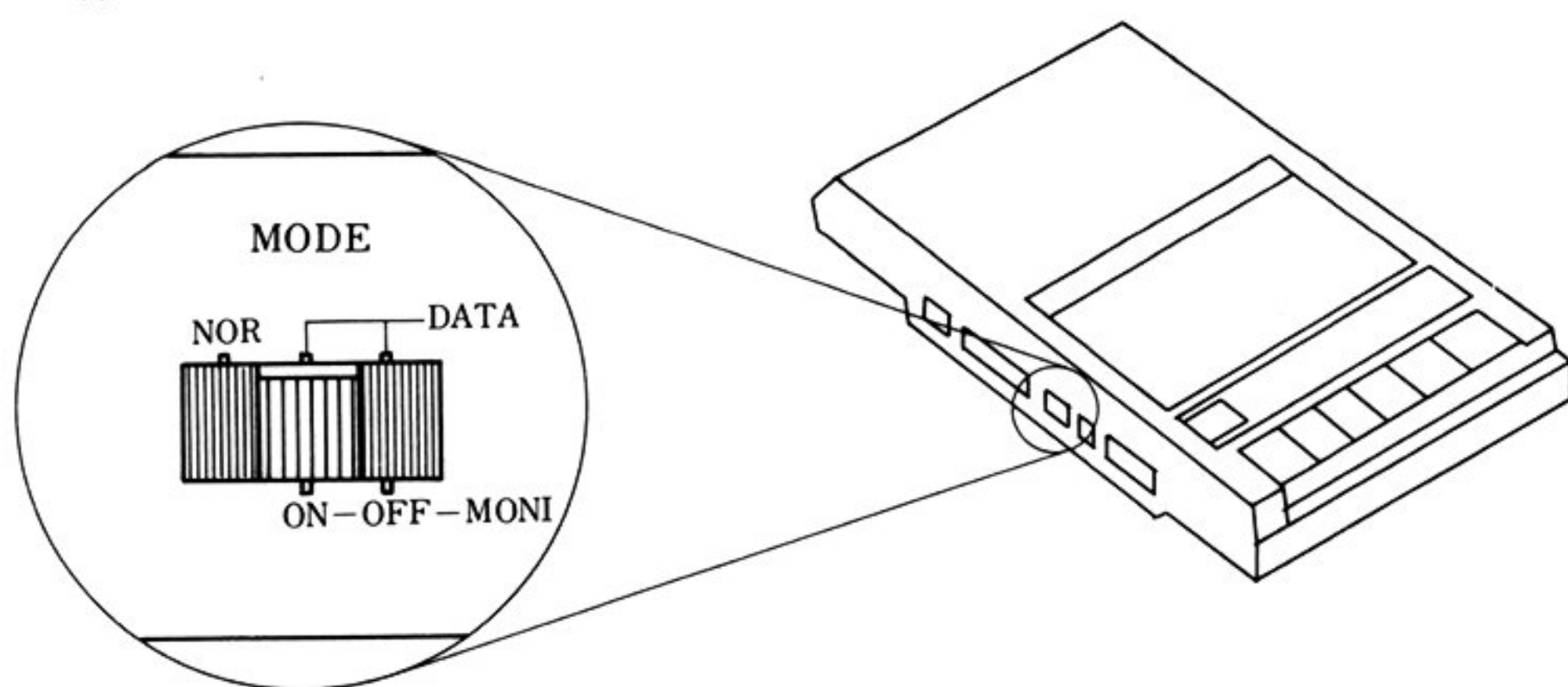
(12) 10と11の操作をプログラムがロード可能になるまで，音質を少しずつ変えながら行ないます．

(13) Found DEMO1とDevice I/O Errorのメッセージがどちらも出力されない場合は，音量が小さすぎますので，音量を上げて下さい．

12.1.2 FMデータレコーダに対する扱い

FMデータレコーダは，モードスイッチを持ち，音楽等の録音再生用と，パーソナルコンピュータに対するモードとに切換えることができます．その他は，家庭用テープレコーダと同じ扱いです．

プログラムのロード・セーブやデータの入出力を行なう場合は，必ずモードスイッチを〝DATA〞側にして行なって下さい．

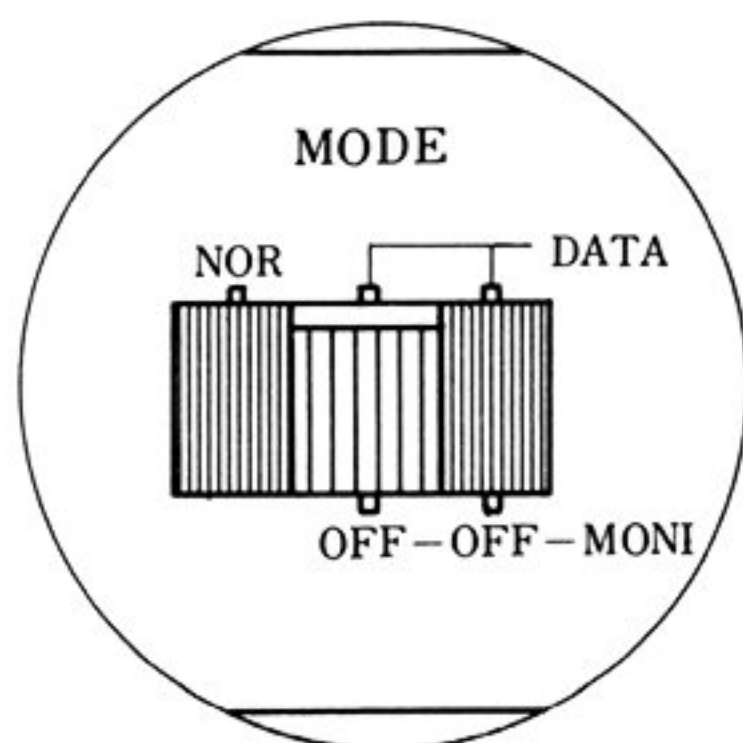
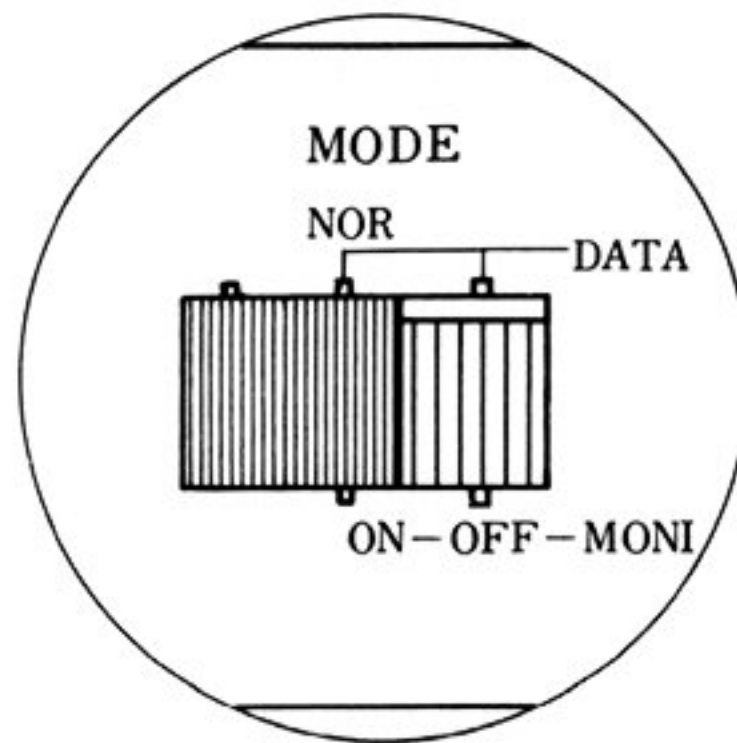
MODE
NOR　DATA
ON－OFF－MONI

モードスイッチの〝DATA〞はさらに，記録されたプログラムやデータを，スピーカからの〝ビー音〞により確認できます．

確認する場合は，〝MONI-ON〞側へ，確認しない場合は，〝MONI-OFF〞側へ設定して下さい．

〝ビー音〞により確認する場合

MODE
NOR　DATA
OFF－OFF－MONI

〝ビー音〞により確認しない場合

MODE
NOR　DATA
ON－OFF－MONI

記録するレベルや，〝ビー音〞の音量は一定レベルに調整されます．

(1) モードスイッチやフェイズコントロールスイッチは，テープの走行中，切換えないで下さい．

(2) 使用後は，必ずカセットテープを取り出し，PLAY ボタンや早送り，巻戻しボタンを押したままの状態で，長時間放置しないで下さい．

(3) リモート機能を使用しない場合は，ポーズ（一時停止）ボタンを利用すると便利です．

12.2 リモートコントロール機能

本体添付のケーブルにより，本体とテープレコーダとを接続しますが，その際，ケーブルのリモート端子（黒色の端子）を使用することにより，カセットテープの走行を ON-OFF させることができます．

この端子を使用しない場合は，PLAY，STOP ボタンにより，テープレコーダの操作を行ないます．

カセットテープが走行している状態を〝ON 状態〟，リモート機能により止まっている状態を〝OFF 状態〟といいます．

F-BASIC 起動時は常に OFF 状態となっています．

(1) OFF 状態から ON 状態への切換え

MOTOR ON ⏎

この入力により，OFF 状態から ON 状態へ切換わり，テープが走行を始めます．

(2) ON 状態から OFF 状態への切換え

MOTOR OFF ⏎

この入力により，ON 状態から OFF 状態へ切換わり，走行していたテープが止まります．

(3) 状態の切換え

MOTOR ⏎

MOTOR コマンドにおいて，ON または OFF の指定を行なわない場合は，ON 状態であれば OFF 状態へ，OFF 状態であれば ON 状態へ切換わります．

リモートコントロール機能は，他の命令においても使われており，その命令開始と共に ON 状態となり，命令が終了すれば OFF 状態となります．

また，命令実行中において，エラーが発生した場合は，OFF 状態となります．

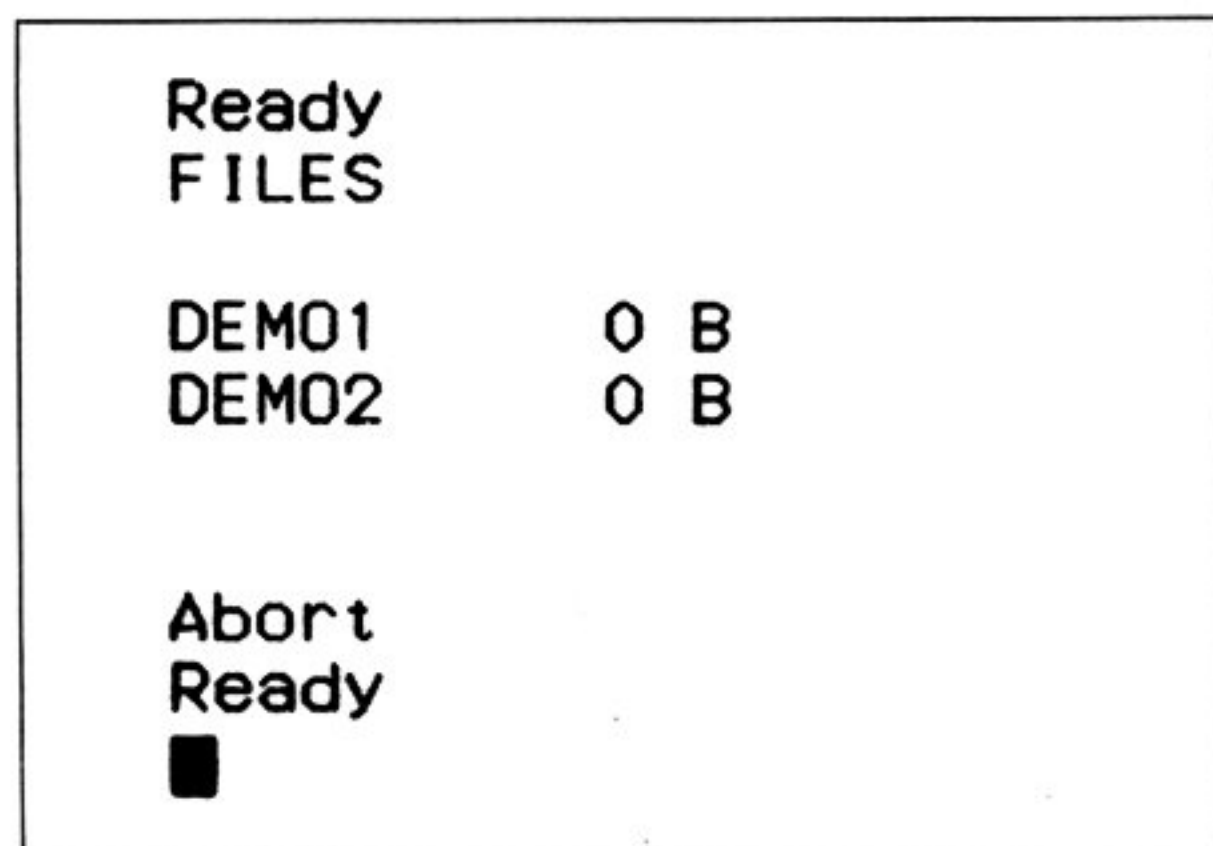
12.3 ファイル名の出力方法 (FILES)

カセットテープに記録されているプログラムファイルや，データファイルのファイル名をディスプレイへ表示させるためのコマンドが，〝FILES（ファイルズ）〟です．

FILES 〝CASO：〟 ⏎

└ROM モードの場合は省略できます

本体に添付のカセットテープを使用した場合の画面表示は以下のようになります.

```
Ready
FILES

DEMO1     0 B
DEMO2     0 B


Abort
Ready
■
```

DEMO1 0 B
- B→ファイルの記録形式
  - B：バイナリ形式
  - A：アスキー形式
- 0→ファイルの種類
  - 0：F-BASICにより作成されたプログラム
  - 1：データファイル
  - 2：機械語プログラムファイル

FILESコマンド実行後, リモート機能はON状態となり, カセットテープ上のファイル名が記録されている場所へくると, ファイル名を読み, ディスプレイへ表示します.

FILESコマンドは, カセットテープが最後まで巻き取られても, 実行を終了しませんので, 終了させる場合は, BREAK を押して下さい.

また, FILESコマンドにより, ファイル名をプリンタへも出力できます.

FILES "CASO:", L ↵

12.4 プログラム ファイルの扱い方

12.4.1 LOADとSAVE

F-BASICにより作成したプログラムを, カセットテープへ記録するための命令が"SAVE (セーブ)"であり, 記録されたプログラムを本体のメモリへ展開する命令が"LOAD (ロード)"です.

LOAD "ファイルディスクリプタ", R

- "ファイルディスクリプタ"→カセットテープに記録されている最初のプログラムをロードする場合は省略できます.
- R→ロード終了後, プログラムの実行を指定します.

本体に添付されているカセットテープを使用し, プログラムのロードを行なう例を示します.

(1) カセットテープをテープレコーダへ入れ，巻き戻しておきます．12.1節のテープレコーダの扱い方を参照し，音量，音質等を調整しておきます．

(2) リモート機能を使用し，OFF状態であれば，PLAYボタンを押します．リモート機能を使用しない場合は，PLAYボタンはまだ押さないで下さい．

(3) 次のように入力します．

ファイル名〝DEMO1〃のプログラムファイルをロードする場合

LOAD 〝CASO：DEMO 1〃 ⏎

カセットテープに記録されているプログラムの最初のファイル名が〝DEMO1〃の場合は，ファイルディスクリプタは省略できます．(ROMモードの場合)

LOAD ⏎

DISKモードで使用する場合は，ファイルディスクリプタのデバイス名は省略できません．

LOAD 〝CASO：〃 ⏎

(4) リモート機能を使用しておれば，カセットテープが走行をはじめます．

リモート機能を使用しない場合，PLAYボタンを押します．

(5) ディスプレイへ〝Searching〃と表示されます．

ファイル名〝DEMO1〃のプログラムファイルの前に，他のプログラムが記録されておれば，そのプログラムのファイル名をディスプレイへ表示し，内容は読みとばします．

読みとばすことを〝Skip(スキップ)〃するといいます．たとえば，〝F-DEMO〃というファイル名を持つプログラムファイルが，〝DEMO1〃プログラムの前に存在したとすれば，ディスプレイへは〝Skip：F-DEMO〃と表示します．

また，指定したプログラム以外のファイルが見つかると，同様にスキップし，そのファイル名をディスプレイへ表示します．

〝TEST〃というファイル名を持つファイルをスキップしたとすれば，ディスプレイへは以下のように表示されます．

Skip：TEST

(6) 次に，プログラムが見つかれば，〝Found：DEMO1〃と表示し，〝Ready〃と表示すれば，プログラムのロードを終了したことを示します．

(7) ロード終了時，リモート機能を使用しておればOFF状態となり，テープの走行は停止します．

リモート機能を使用しない場合は，〝Ready〃が表示された後STOPボタンを押し，テープを止めます．

(8) ロード実行中〝Device I/O Error〃等のエラー表示，または，何も表示しないまま，テープが巻き取りを終了した場合は，もう一度12.1節を参照し，調整を行なって下さい．

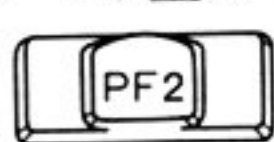
プログラムをロード終了後実行する場合は，以下のように入力して下さい．

LOAD "CASO：DEMO1", R ⏎

次に，メモリ上のプログラムのセーブ方法を，ロード後の"DEMO1"プログラムを使用し，説明します．

SAVE "ファイルディスクリプタ", A
└→ROMモード使用の場合は，デバイス名のみ省略できます． └→アスキー形式によりセーブを行なう指定です．指定がなければ，バイナリ形式によりセーブします．

ロードを行なったメモリ上の"DEMO1"プログラムを，他のカセットテープへセーブします．

(1) メモリ上のプログラムを確認します．以下のように入力して下さい．

PF2 （または LIST ⏎ ）

ディスプレイへプログラムのリストが表示されることを確認して下さい．リストが表示されない場合は，再度，プログラムをロードして下さい．

(2) テープレコーダへ，セーブするためのカセットテープを入れ巻き戻します．ロードした時と同じ音量，音質に調整します．

カセットテープにおいては，以前記録してあったプログラムの上へ，次のプログラム等を記録すると，以前のプログラムは消えてしまいますので注意して下さい．

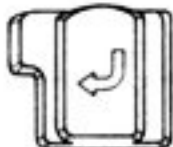
(3) リモート機能を使用する場合は，PLAYボタンとREC（録音）ボタンを同時に押し，録音状態としておきます．

(4) リモート機能を使用する場合は，MOTORコマンドにより，テープのクリアフィーダと呼ばれる録音できない部分を送っておきます．クリアフィーダは磁性体（かっ色）が塗られてないことより，透明または白色となっております．送り終れば，再度，MOTORコマンドにより，OFF状態にしておきます．

リモート機能を使用しない場合も，PLAYボタンとRECボタンを同時に押し，クリアフィーダを送り終ればSTOPボタンを押し，テープの走行を停止させます．

カセットテープの途中，または，前にプログラムがある場合は，必ず前のプログラムとの間をあけるようにして下さい．

プログラムファイルを読みとばす場合は，次の入力を行ないます．

SKIPF "ファイルディスクリプタ" ⏎

ROMモード使用時は，ファイルディスクリプタのデバイス名を省略でき，ファイル名を省略した場合は，先頭のファイルを読みとばします．

このコマンド実行後，録音状態において，少しテープを進めておき，前のプログラムとの間隔を

あけておきます.

(5) 次のように入力して，セーブを行ないます.

SAVE "CASO：DEMO"

リモート機能を使用しない場合は，[↵]を入力する前に，PLAYボタンとRECボタンを同時に押し，録音状態としておきます.

(6) "Ready" とディスプレイへ表示すれば，プログラムのセーブが終了したことを示します.

(7) セーブ終了時，リモート機能を使用しておれば，OFF状態となり，テープの走行は停止します.

リモート機能を使用しない場合は，"Ready" が表示された後，STOPボタンを押し，テープを止めます.

12.4.2 LOADMとSAVEM

機械語により作成されたプログラムをカセットテープへ記録する命令が，SAVEM(セーブエム)コマンドであり，カセットテープに記録されている機械語プログラムをメモリ上へ展開する命令が，LOADM(ロードエム) コマンドです.

カセットテープおよびテープレコーダに対する扱いは，12. 4. 1 LOADとSAVEの場合と同じです.

SAVEM "ファイルディスクリプタ", 開始番地, 終了番地, 入口番地

- ファイルディスクリプタ → ROMモードの場合デバイス名は省略できます.
- 開始番地 → 記録を開始するメモリの番地を指定します.
- 終了番地 → 記録を終了するメモリの番地を指定します.
- 入口番地 → 機械語プログラムの実行番地を指定します.

開始番地と終了番地により，メモリ上の記録する範囲を指定します. 入口番地により，機械語プログラムの実行番地を指定します.

たとえば，メモリ上の6000番地（16進）より63FF番地（16進）までの機械語プログラムがあり，実行番地は60D0（16進）番地であった場合の記述方法を示します.

開始番地 ………… &H6000 （&HはF-BASICにおいて16進数を示す場合記述します）
終了番地 ………… &H63FF
入口番地 ………… &H60D0

ファイル名を "MACHINE" とします.

SAVEM "CASO：MACHINE", &H6000, &H63FF, &H60D0

開始番地，終了番地，入口番地は必ず指定します. 指定のない場合は，エラー（Syntax Error）となります.

LOADM "ファイルディスクリプタ", オフセット値, R

- ファイルディスクリプタ→カセットテープに記録されている最初のプログラムをロードする場合は，省略できます．
- オフセット値→SAVEMコマンドにおいて指定した開始番地に加えられ，その番地よりメモリへ展開されます．
- R→プログラムの実行を指定します．

本体内メモリへ展開する番地は，オフセット値を記述しない場合，SAVEMコマンドにより，指定した開始番地より，終了番地となります．これは，SAVEMコマンドにより指定した番地が，カセットテープへ記録されるためです．

LOADM "CASO：MACHINE" ⏎

ROMモードを使用の際，カセットテープの先頭にあるプログラムをロードする場合は，ファイルディスクリプタを省略できます．

LOADM ⏎

オフセット値の指定を行なうと，プログラムをメモリへ展開する番地を変更できます．

SAVEMコマンドにより指定した開始番地と終了番地へオフセット値が加算され，その番地をもとにプログラムがメモリへ展開されます．

SAVEM "CASO：MACHINE", &H6000, &H63FF, &H60D0

上の例は，SAVEMコマンドにおいて示したものですが，7000番地（16進）よりプログラムを展開させる場合のオフセット値は，

&H 7000－&H 6000 = &H 1000 となり，以下のように記述します．

LOADM "CASO：MACHINE", &H1000

この記述により以下のように，番地が変更されます．

開始番地　&H6000＋&H1000 = &H7000

終了番地　&H63FF＋&H1000 = &H73FF

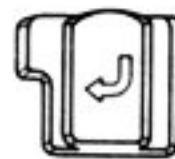
入口番地　&H60D0＋&H1000 = &H70D0

その他，オフセット値がマイナスであれば，その分各番地は減算されます．オフセット値により変更された番地が，FFFF番地（16進）を越える場合は，&H 10000を引いた値が変更後の番地となり，変更された番地がマイナスの値となれば，その値を &H 10000より引いた番地となります．

機械語プログラムをロードした後，入口番地より，実行させるためには，以下のように入力します．

LOADM "CASO：MACHINE",, R ⏎

この場合も，オフセット値を指定できます．オフセット値を &H 1000とすると，以下のように入力します．

LOADM "CASO : MACHINE", &H1000, R ⏎

また，機械語プログラムを，F-BASICにより作成されたプログラムのサブルーチンとして使用する等の場合は，F-BASICプログラムにより使用する領域を制限しておかなければなりません．

CLEAR（クリア）命令を使用します．

CLEAR [文字領域の大きさ[, F-BASICプログラムの使用する上限のメモリ番地]

たとえば，SAVEMコマンドにおいて示した例では，機械語プログラムを6000番地（16進）よりカセットテープへセーブしております．このプログラムをそのまま6000番地（16進）へロードするためには，以下のように記述します．

文字領域の大きさを300バイトとします．

CLEAR 300, &H 5FFF

また，F-BASICにより作成されたプログラム中に，記述しておけば，プログラム実行中に処理されます．

```
10  CLS
20  CLEAR  300, &H 5FFF
30  LOADM "CASO : MACHNE"
40
 :
500 END
```

12.4.3 ロード，セーブに関する注意事項

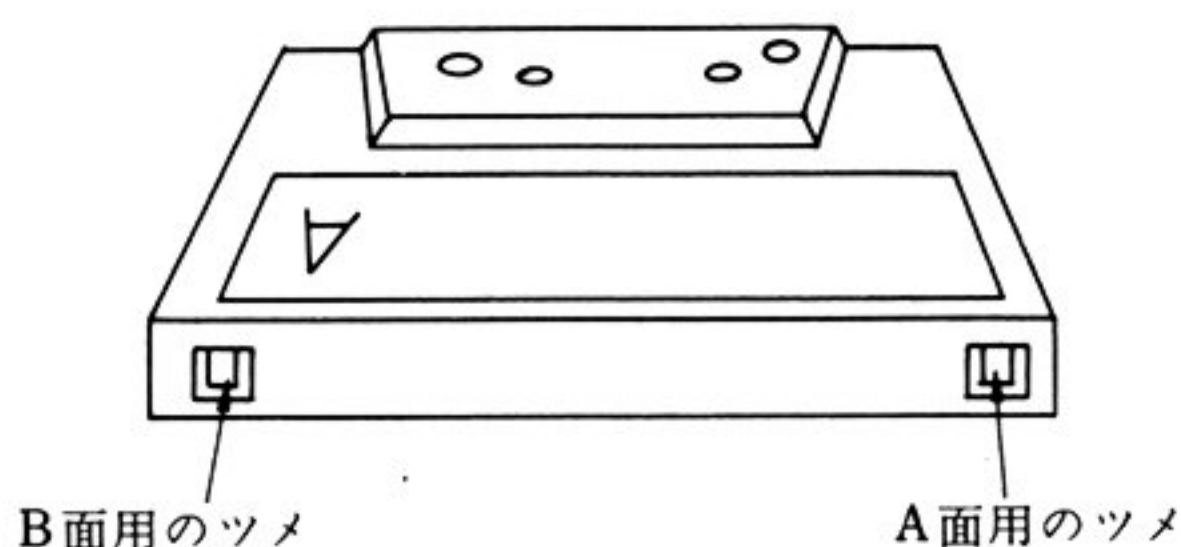
(1) プログラムのロード，セーブを行なう際には，必ず，接続，音量，音質が適当であるかを確認して下さい．

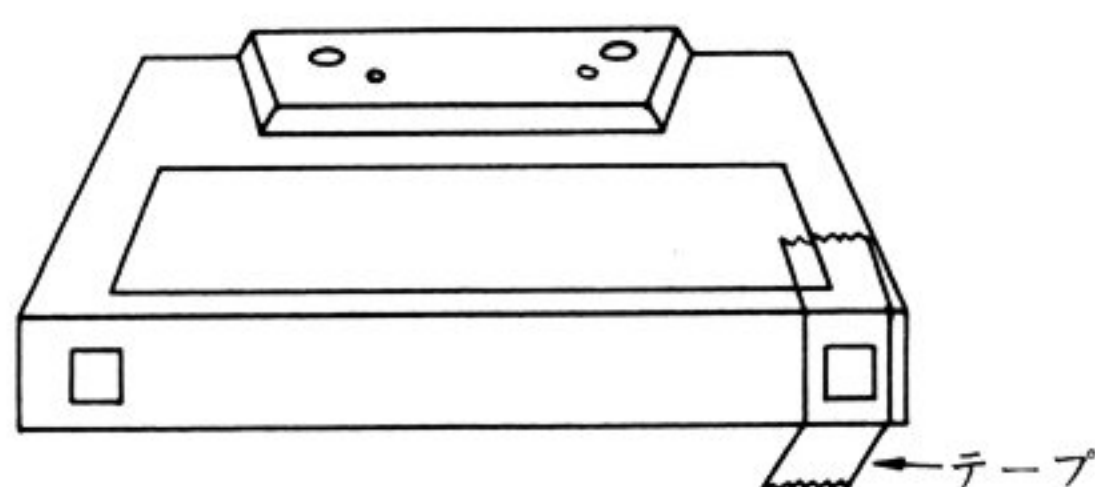
(2) カセットテープ上のプログラム間は，必ず間隔をあけるようにして下さい．プログラムが重なってしまうとロードできなくなります．

SKIPF命令によりプログラムを読みとばし，その後にプログラム等をセーブする場合も，録音状態にしてカセットを少し進めておき，間隔をあけるようにして下さい．

(3) 行番号を付け，プログラム中においてロード，セーブを行なう場合は，"Searching"および"Skip :" のメッセージは出力されません．

(4) 大切なプログラムをセーブした場合，間違って消さないためには，カセットテープのツメを折っておく必要があります．

B面用のツメ　　A面用のツメ

ツメを折った後テープ等を貼ることにより，再度録音可能となります．

←テープ

12.5 チェックサムの照合

カセットテープへプログラムやデータを記録する際，データをある範囲で区切り，区切ったデータを，8ビットの範囲で加算します．8ビットを越えたもの（オーバフロー）は，切り捨ててしまい，加算し終った8ビットを〝データ″として記録します．このデータを〝チェックサム″といいます．

このチェックサムと，カセットテープへ記録されたデータとを照合する命令が〝LOAD？″（ロードクエスション）命令です．この命令は，カセットテープのみ実行可能な命令です．

また，LOAD，LOADM命令においても，チェックサムの照合を行ないます．

LOAD？[〝[CASO：]ファイル名″]
└→ROMモード使用時は省略できます

本体メモリ上のプログラムとは関係なく，カセットテープ上のデータとチェックサムの照合を行ないます．

照合した結果が一致しない場合は，エラー（Device I/O Error）が発生します．

他のデバイスに対し実行できません．他のデバイスを指定するとエラー（illegal function call）が発生します．

添付のカセットテープの〝DEMO1″プログラムについてチェックサムの照合を行なう場合の入力方法を示します．

LOAD？ 〝CASO：DEMO1″ ⏎

カセットテープの先頭にあるプログラムファイルのチェックサムを照合する場合は，ファイルディスクリプタを省略できます．

LOAD ? ⏎

指定外のファイルを読みとばした場合，そのファイル名を，ディスプレイへ出力します．

Skip：ファイル名

また，指定したファイルが見つかれば，メッセージを出力します．

Found　ファイル名

12.6 データ ファイルの扱い方

カセットテープにおけるデータファイルは，シーケンシャルファイルとなります．

データを記録する際は，リモート機能を使用すると，カセットテープに無駄が生じません．以下の説明はリモート機能を使用しているものとして行ないます．

また，リモート機能を使用しない場合，あるデータを読み，そのデータを基に，処理を行なった後，さらに次のデータを読もうとしたとき，そのデータを記録した場所はすでに巻きとられ，データがなくなっていたり，違うデータであったりしますので注意して下さい．

12.6.1 データ ファイルの作成

カセットテープへデータを記録するための簡単なプログラムを示し，必要な命令や，データの記録形式等を説明します．

データファイルを作成するためには，以下の処理が必要です．

(1) カセットテープに対する出力のためのファイルを開く．

OPEN 命令…… OPEN "O"，#ファイル番号，"CASO：ファイル名"

(2) 記録するデータを入力する．(キーボードより)

INPUT 命令…… INPUT "メッセージ，変数名"

(3) ファイルへデータを記録する．

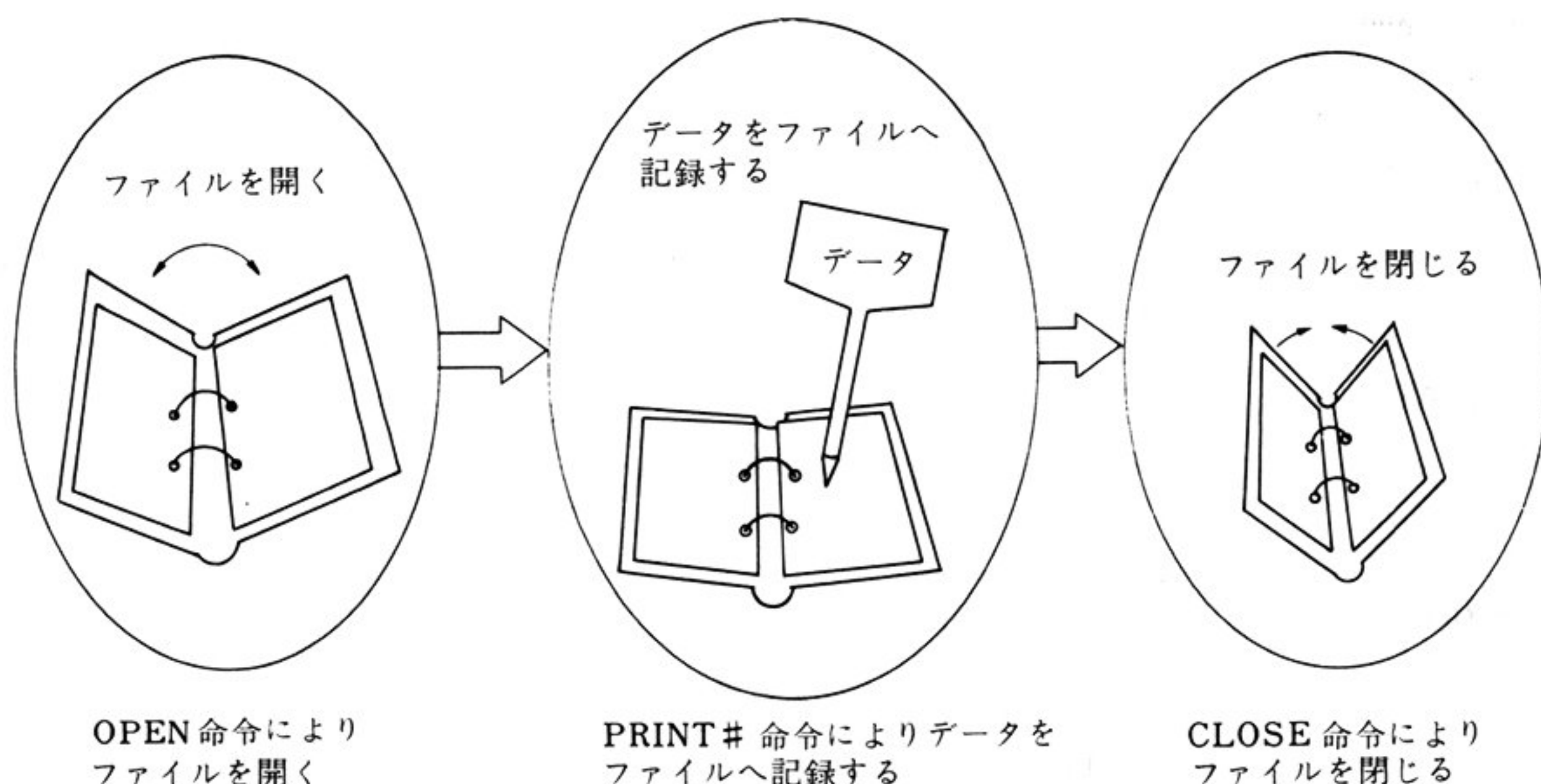
PRINT #命令…… PRINT #ファイル番号，変数名

(4) ファイルを閉じる．

CLOSE 命令…… CLOSE

OPEN命令により
ファイルを開く

PRINT# 命令によりデータを
ファイルへ記録する

CLOSE命令により
ファイルを閉じる

以下にプログラム例を示します.

このプログラムは, 食堂の店名, 電話番号, 品名と価格を入力し, それらをカセットテープへデータとして記録するためのプログラム例です.

```
100 CLS:WIDTH40,20
110 LOCATE 0,3
120 PRINT"カセット テープ ハ OK デスカ?"
130 PRINT"OK ナラ Y    "
140 IF INKEY$<>"Y" THEN GOTO 140
150 CLS:Y=3
160 INPUT"ファイルメイ ヲ ニュウリョクシテクダサイ        ,NA$:CLS
170 LOCATE 0,5:PRINT"イマ ファイル ヲ ヒライテイマス"
180 OPEN"O",#1,"CASO:"+NA$
190 CLS
200 LOCATE 0,0:PRINT"ファイルメイ--------"+NA$
210 LOCATE 8,2:PRINT"テンメイ"SPC(12)"デンワバンゴウ"
220 LOCATE 0,18:INPUT "テンメイ ヲ ニュウリョク       ",A$
230 LOCATE 0,18:PRINT SPC(39)
240 LOCATE 0,18:INPUT"デンワバンゴウ ノ ニュウリョク       ",B$
250 GOSUB 1000
260 Y=Y+2
500 LOCATE 8,5:PRINT"ヒンメイ"SPC(12)"カカク"
510 LOCATE 0,18:INPUT"ヒンメイ ノ ニュウリョク       ",A$
520 LOCATE 0,18:PRINT SPC(39)
530 LOCATE 0,18:INPUT"カカク ノ ニュウリョク       ",B$
540 GOSUB 1000
550 LOCATE 0,18:PRINT"ニュウリョク ハ オワリデスカ?  オワリ ナラ E    ":E$=INPUT$(1)
560 LOCATE 0,18:PRINT SPC(39)
570 IF E$<>"E" THENGOTO 510
580 PRINT"ニュウリョク ノ ショリ ハ オワリ デス"
590 CLOSE:END
1000 LOCATE 0,18:PRINT SPACE$(39)
1010 IF Y<17 THEN GOTO 1030
1020 Y=6:LINE (0,60)-(639,170),PRESET,,BF
1030 LOCATE 6,Y:PRINT USING"&                  &    &                  &";A$,B$
1040 PRINT#1,A$,B$
1050 Y=Y+1
1060 RETURN
```

プログラム説明

以下に前述したプログラムの説明を述べます．

100　CLS : WIDTH 40, 20

画面を CLS 命令によりクリアして，WIDTH 命令により，画面のキャラクタ表示を横 40 桁，縦 20 行に設定します．

110　LOCATE　0, 3

カーソルを縦 3 行目の左端に指定します．LOCATE 命令は，カーソルの位置を指定します．

120　PRINT "カセット　テープ　ハ　OK　デスカ？"

カセットテープとテープレコーダに対する準備を行なうように，メッセージを出力します．メッセージは，行番号 110 の LOCATE 命令において指定された位置より画面へ表示します．

130　PRINT "OK　ナラ　Y"

カセットテープの準備ができれば [Y] を押すようメッセージを出力します．

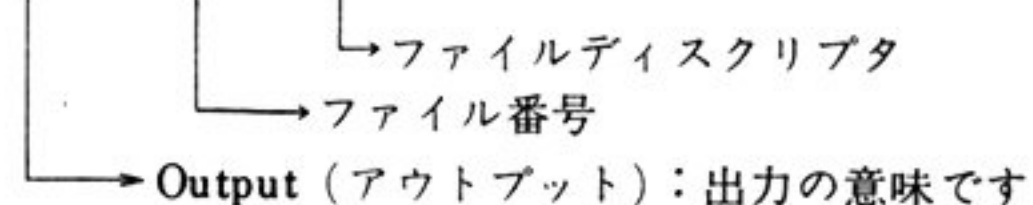
140　IF　INKEY$ <>　"Y" THEN　GOTO　140

押されているキーの文字を読み取る命令が INKEY$ 命令で，この行では押されているキーが，[Y] 以外であれば，[Y] が押されるまで止まっています．[Y] のキーが押されると，次の行へ実行を移します．

150　CLS : Y=3

画面をクリアし，Y という数値変数へ 3 を設定します．Y という変数は，後に LOCATE 命令における縦の行数を与えます．

160　INPUT "ファイルメイ　ヲ　ニュウリョクシテクダサイ　", NA$: CLS

カセットテープに記録するためのファイル名を入力するようメッセージを出力します．ファイル名は，NA$ という文字変数へ記憶されます．

170　LOCATE　0,5 : PRINT "イマ　ファイル　ヲ　ヒライテイマス"

次行の OPEN 文に対するメッセージを画面に出力します．

180　OPEN "O", #1, "CASO : " +NA$

カセットテープに対するデータファイル（シーケンシャルファイル）を開き，バッファを準備します．

```
OPEN "O", #1, "CASO:" +NA$
      |    |   └→ファイルディスクリプタ
      |    └→ファイル番号
      └→Output（アウトプット）：出力の意味です
```

ファイル番号は，1～16 まで選択できますが，1 つのファイルに対するファイル番号は，

すべてOPEN命令により，指定された番号を用います．

ファイルディスクリプタのファイル名は，行番号160のINPUT命令により，NA＄という文字変数へ記憶したものが用いられます．

たとえば，行番号160において〝DATA〟と入力したとすれば，ファイルディスクリプタは，〝CASO：DATA〟となります．

190 CLS

画面をクリアしています．

200 LOCATE 0, 0：PRINT 〝ファイルメイ………〟＋NA$

画面の左上へファイル名を表示させます．

210 LOCATE 8, 2：PRINT 〝テンメイ〟 SPC(12) 〝デンワバンゴウ〟

画面の2行目へ店名と電話番号という項目名を表示します．

SPC命令は，SPACE(スペース：空白）を12個表示することを示します．

220 LOCATE 0, 18：INPUT 〝テンメイ ヲ ニュウリョク〟, A$

画面の18行へ店名を入力するようメッセージを出力し，入力された店名は，A＄という文字変数へ記憶されます．

INPUT命令は，データの最後に[↵]を押すことにより入力されます．

230 LOCATE 0, 18：PRINT SPC(39)

行番号200において出力されたメッセージを空白により消しています．

240 LOCATE 0, 18：INPUT 〝デンワバンゴウ ノ ニュウリョク〟, B$

画面の18行へ電話番号を入力するようメッセージを出力し，入力された店名は，B＄という文字変数へ設定されます．

250 GOSUB 1000

実行を行番号1000からのサブルーチンへ移します．サブルーチンの実行終了後，RETURN文により，次行へ実行が移ります．

260 Y＝Y＋2

数値変数〝Y〟の値へ2を加えます．

500 LOCATE 8, 5：PRINT 〝ヒンメイ〟 SPC(12) 〝カカク〟

画面の5行目へ品名と価格という項目名を表示します．

510 LOCATE 0, 18：INPUT 〝ヒンメイ ノ ニュウリョク 〟, A$

画面の18行目へ品名を入力するようメッセージを出力し，入力された品名は，A＄という文字変数へ設定されます．

520 LOCATE 0, 18：PRINT SPC(39)

行番号510において出力されたメッセージを空白により消しています．

530　LOCATE　0, 18 : INPUT "カカク　ノ　ニュウリョク　", B$

画面の18行目へ価格を入力するよう，メッセージを出力し，入力された価格は，B$という文字変数へ設定されます．

540　GOSUB　1000

実行を行番号1000からのサブルーチンへ移します．

550　LOCATE 0,18 : PRINT "ニュウリョク　ハ　オワリデスカ？　オワリ　ナラ　E" : E$=INPUT$(1)

画面の18行目へ入力が終りなら [E] を押すよう，メッセージを出力します．

INPUT$文は，"( )"（かっこ）により指定された文字数を，入力できます．この場合の指定は(1)ですので，E$へキーボードより1文字入力し，設定されます．

560　LOCATE　0, 18 : PRINT　SPC(39)

行番号550のメッセージを空白により消しています．

570　IF　E$ <> "E"　TNEN　GOTO　510

行番号550において入力された文字が，処理終了を示す"E"であるかの判定を行なっています．"E"以外の文字であれば，行番号510へ処理を移します．

580　PRINT "ニュウリョク　ノ　ショリ　ハ　オワリ　デス"

処理の終了を示すメッセージを出力します．

590　CLOSE : END

ファイルをCLOSE命令により閉じ，END命令により実行の終了を指定します．

1000　LOCATE　0, 18 : PRINT　SPACE$(39)

この行以降は，サブルーチンとしています．"GOSUB　1000"という指定により，この行へ実行が移り，行番号1060のRETURN命令により，GOSUB命令の次の命令を実行します．

この行番号は，画面の18行目のメッセージを空白により消去します．

1010　IF　Y <17　THEN　GOTO　1030

数値変数"Y"の値が17に満たない場合は，行番号1030へ実行を移します．

"Y"は品名と体格を画面へ表示させるための行数を指定します．

1020　Y=6 : LINE　(0, 60)−(639, 170),　PRESET, , BF

行番号1010において行数である"Y"が17以上であれば，この行番号により，行数を6と設定し，LINE命令により画面へ表示している品名と価格を消去します．

LINE　(0, 60)−(639, 170),　PRESET,　,　BF

- (0, 60)−(639, 170) → 画面の座標を指定します．
- PRESET → 機能を指定します．この場合は，消去を意味します．
- BF → 座標により示された対角線の箱を塗りつぶす指定です．

1030 LOCATE 6, Y : PRINT USING "&　　& &　　&" ; A$, B$

画面の "Y" で指定された行へ，文字変数 A$ と B$ の内容を PRINT USING 命令により表示します．

PRINT USING "&　　& &　　&"; A$, B$

→この間は，空白が出力されます．

→出力する文字変数

→&(ampersand：アンパサント) 記号により囲まれた空白の数に2を加えた数の文字を，文字変数 A$ と B$ の内容より表示します．

1040 PRINT #1, A$, B$

ファイルへ，文字変数 A$ と B$ の内容を出力します．

PRINT #1, A$, B$

→ファイル番号　→変数名

カセットテープに対する文字変数の区切りは，"；(セミコロン)" であっても，"，(コンマ)" であっても CR (キャリッジリターン) で区切られます．

(例) このように区切ったとしてもファイルにおけるデータの区切りは変わりません．

PRINT #1, A$; B$;

1050 Y=Y+1

画面へ品名と価格を出力する際の行数指定である数値変数 "Y" の価を1増加します．

1060 RETURN

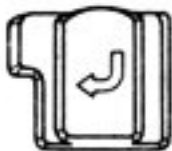
サブルーチンの処理終了を示します．このサブルーチンへ実行を移した GOSUB 文の次の命令を実行します．

例に示したプログラムの実行例

プログラムを実行します．

RUN [↵]

```
Ready
RUN
```

画面へ，カセットテープを準備するように，メッセージが出力されます．このメッセージにより，カセットテープをテープレコーダへ入れ，録音状態にします．

カセットテープは，リモート機能により停止しています．

準備完了であれば，[Y] を押します．

カセット テープ ハ OK デスカ?
OK ナラ Y

ファイル名を入力するようにメッセージを出力します．

［例］

エフエムヤ[↵]と入力します．

ファイル名の後には必ず[↵]を入力します．

ファイルメイ ヲ ニュウリョクシテクダサイ　エフエムヤ

ファイル名を入力後，メッセージを出力して，カセットテープが回り始めます．

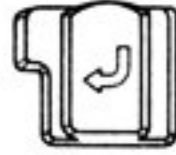
カセットテープへファイル名を記録して，カセットテープは停止します．

ここで，ファイルとバッファの準備が完了します．

イマ ファイル ヲ ヒライテイマス

ファイル名，項目を出力し，店名を入力するようメッセージを出力し，カーソルが点滅します．

例としてファイル名と同じ名前を入力します．

エフエムヤ ⏎

```
ファイルメイ－－－－－－－－エフエムヤ
        テンメイ              デンワバンゴウ

テンメイ ヲ ニュウリョク     エフエムヤ
```

次に電話番号を入力するようメッセージを出力し，カーソルが点滅します．

[例]

00-111-2222 ⏎

と入力します．

```
ファイルメイ－－－－－－－－エフエムヤ
        テンメイ              デンワバンゴウ

デンワバンゴウ ノ ニュウリョク     00-111-2222
```

店名と電話番号を出力した後，次の項目，および品名を入力するようメッセージを出力します．入力した店名および電話番号はすでにファイルに対するバッファへ蓄えられています．

[例]

テンプラテイショク ⏎

と入力します．

```
ファイルメイ－－－－－－－－エフエムヤ
        テンメイ              デンワバンゴウ
      エフエムヤ             00-111-2222
        ヒンメイ              カカク

ヒンメイ ノ ニュウリョク     テンプラテイショク
```

価格を入力するようメッセージを出力します．テンキーより入力すると便利です．

［例］

1200 ⏎

と入力します．これは，1,200円を意味します．

```
ファイルメイー－－－－－－－エフエムヤ

テンメイ                 デンワバンゴウ
エフエムヤ                00－111－2222

ヒンメイ                 カカク

カカク ノ ニュウリョク        1200
```

入力した品名と価格を出力し，入力が終了したかどうかのメッセージを出力します．

さらに入力する場合は，〝E〟以外のキーを入力します．

［例］

⏎ を押します．

品名と価格のデータは，店名と同じようにバッファへ蓄えられます．

```
ファイルメイー－－－－－－－エフエムヤ

テンメイ                 デンワバンゴウ
エフエムヤ                00－111－2222

ヒンメイ                 カカク
テンプラテイショク            1200

ニュウリョク ハ オワリデスカ? オワリ ナラ E
```

再度，品名を入力するよう，メッセージを出力します．

以下，入力すると，価格，終了のメッセージを出力しますので，前述したように入力していきます．

```
ファイルメイー－－－－－－－エフエムヤ

テンメイ                 デンワバンゴウ
エフエムヤ                00－111－2222

ヒンメイ                 カカク
テンプラテイショク            1200

ヒンメイ ノ ニュウリョク
```

```
ファイルメイーーーーーーーーエフエムヤ

    テンメイ              デンワバンゴウ
  エフエムヤ              00-111-2222

    ヒンメイ              カカク
  テンプラテイショク      1200
  トロロテイショク        1000
  ソバテイショク          600
  ソバベントウ            800
  テンザラ                1000
  テンドンジョウ          900
  テンドンナミ            700
  カツドンジョウ          900
  カツドンナミ            600
  オヤコドンジョウ        900
  オヤコドンナミ          600

ニュウリョク ハ オワリデスカ? オワリ ナラ E
```

この画面において，次の品名と価格を入力することにより，今までの品名と価格を消して，入力した品名と価格を出力します．

画面へは，上図のように最大11行出力します．

品名と価格の入力を次々と行なっていると，カセットテープが回り始め，記録を開始します．これは，バッファが一杯になったため，カセットテープへ記録し，バッファを空にしようとするためです．

バッファは255バイト分データを蓄えることができます．F-BASICは，バッファが一杯になるか，CLOSE命令を実行するまで記録を行ないません．

例のプログラムにおいては，店名，電話番号，品名，価格のデータが255バイト分になれば，カセットテープへ記録します．このとき，データ間の区切りとして，CR（キャリッジ・リターン）が設定され，CRも1文字分に数えられます．

データ入力の処理を終る場合は，「ニュウリョク　ハ　オワリデスカ？」というメッセージを出力した後，[E] を押します．

```
ファイルメイーーーーーーーーエフエムヤ

    テンメイ              デンワバンゴウ
  エフエムヤ              00-111-2222

    ヒンメイ              カカク
  テントジドン            600

ニュウリョク ハ オワリデスカ? オワリ ナラ E
```

CLOSE 命令により，バッファ内のデータをすべてカセットテープへ記録した後ファイルを閉じます．

画面へ処理終了したことのメッセージを出力し，コマンドモード（Readyを表示し，カーソルが点滅している状態です）へ戻ります．

```
ファイルメイ－－－－－－－－エフエムヤ

          テンメイ                デンワバンゴウ

          ヒンメイ                カカク
         テントジドン              600


ニュウリョク  ノ  ショリ  ハ  オワリ  デス

Ready
```

12.6.2 データファイルからのデータ入力

カセットテープへデータファイルの作成が終れば，次にそのファイルより，データを読む必要があります．

12.6.1 データファイルの作成において説明に使用したプログラムにより作成されたデータファイルを用いて，データを読み，画面へ出力するためのプログラム例を示し，説明します．

データファイルよりデータを読むためには，以下の処理が必要です．

(1) カセットテープのデータファイルより，データを読むためのファイルを開く．

OPEN 命令……OPEN "I", #ファイル番号, "CAS0：ファイル名"

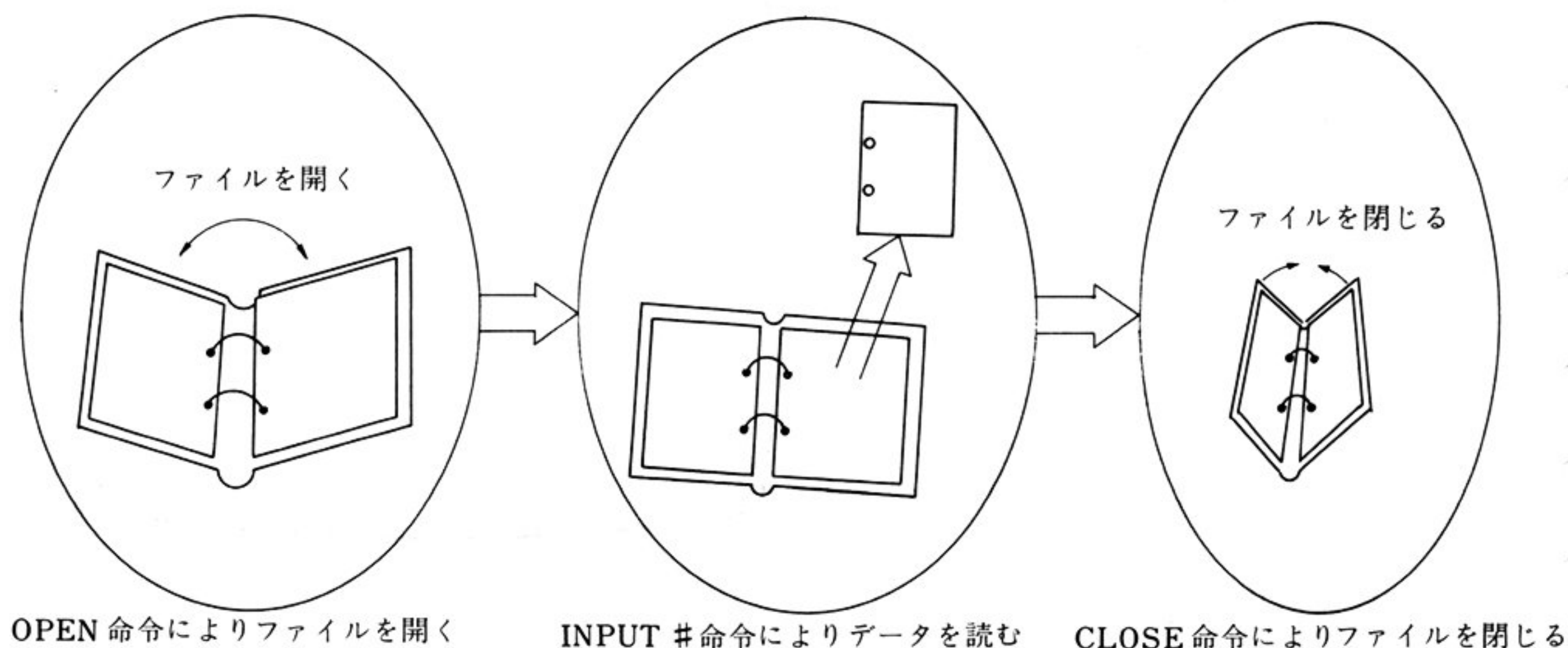
(2) データファイルより，データを読む．

INPUT #命令……INPUT　#ファイル番号, 変数名

(3) ファイルを閉じる．

CLOSE 命令……CLOSE

ファイルを開く　　ファイルを閉じる

OPEN 命令によりファイルを開く　　INPUT #命令によりデータを読む　　CLOSE 命令によりファイルを閉じる

以下にプログラム例を示します.

このプログラムは，12.6.1 データファイルの作成において説明したプログラムにより作成したデータファイルを読み，ディスプレイへ，データを出力するためのプログラム例です.

```
10 CLS:WIDTH40,20
20 LOCATE 0,3
30 PRINT"ｶｾｯﾄ ﾃｰﾌﾟ ﾊ OK ﾃﾞｽｶ?"
40 PRINT"OK ﾅﾗ Y   "
50 IF INKEY$<>"Y" THEN GOTO 50
60 CLS:Y=3
70 INPUT"ﾌｧｲﾙﾒｲ ｦ ﾆｭｳﾘｮｸｼﾃｸﾀﾞｻｲ    ",NA$:CLS
80 LOCATE 0,5:PRINT"ｲﾏ ﾌｧｲﾙ ｦ ｻｶﾞｼﾃｲﾏｽ"
90 OPEN"I",#1,"CASO:"+NA$
100 CONSOLE 6,0
110 LOCATE 0,0:PRINT"ﾌｧｲﾙﾒｲ--------"+NA$
120 LOCATE 8,2:PRINT"ﾃﾝﾒｲ"SPC(12)"ﾃﾞﾝﾜﾊﾞﾝｺﾞｳ"
130 INPUT#1,A$,B$
140 LOCATE 6,3:PRINT USING"&           &   &            &";A$,B$
150 LOCATE 8,5:PRINT"ﾋﾝﾒｲ"SPC(12)"ｶｶｸ";
160 LOCATE 0,6
170 INPUT#1,A$,B$
180 PRINT SPC(6);:PRINT USING"&           &   &            &";A$,B$
200 IF EOF(1) THEN GOTO 210 ELSE 170
210 PRINT"ﾃﾞｰﾀ ｦ ｽﾍﾞﾃ ﾖﾐｵﾜﾘﾏｼﾀ"
220 CLOSE
230 PRINT"ｶﾞﾒﾝ ｦ ｹｽﾅﾗ Y"
240 IF INKEY$<>"Y" THEN 240
250 WIDTH 40,20
260 END
```

プログラム説明

以下に，前述したプログラムの説明を述べます.

100 CLS : WIDTH 40, 20

画面を CLS 命令によりクリアして，WIDTH 命令により，画面のキャラクタ表示を横 40 桁，縦 20 行に設定します.

110 LOCATE 0, 3

カーソルを，縦 3 行目の左端に指定します. LOCATE 命令は，カーソルの位置を指定します.

120 PRINT "カセットテープ ハ OK デスカ？"

カセットテープとテープレコーダに対する準備を行なうように，メッセージを出力します. メッセージは，行番号 110 の LOCATE 命令において指定された位置より画面へ出力します.

130 PRINT "OK ナラ Y"

カセットテープの準備ができれば [Y] を押すようメッセージを出力します.

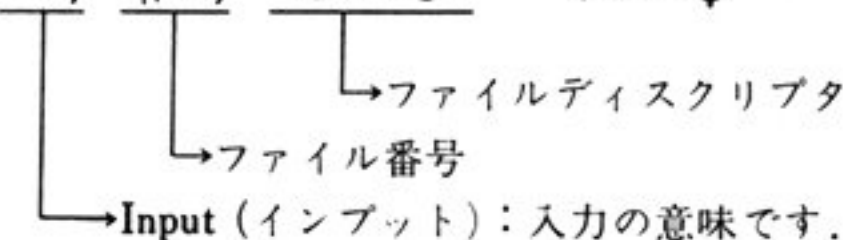
140　IF　INKEY$ <> "Y"　THEN　GOTO　140

押されているキーの文字を読取る命令がINKEY$命令です．この行では，押されているキーが，[Y]以外であれば，[Y]が押されるまで止まっています．[Y]が押されると次の行番号へ実行を移します．

150　CLS：Y=3

画面をクリアして，Yという数値へ3を設定します．

160　INPUT "ファイルメイ　ヲ　ニュウリョクシテクダサイ　"，NA$：CLS

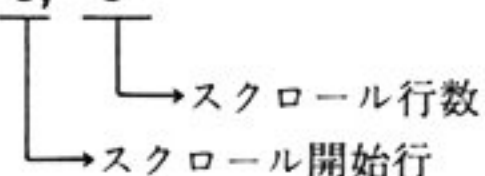
カセットテープに記録されているデータファイルのファイル名を入力します．ファイル名は，NA$という文字変数へ記憶されます．

170　LOCATE　0，5：PRINT "イマ　ファイル　ヲ　ヒライテイマス"

次行のOPEN文に対するメッセージを画面に出力します．

180　OPEN "I"，#1，"CASO：" +NA$

カセットテープよりデータ入力用のデータファイルを開き，バッファを準備します．

```
OPEN "I", #1, "CASO:" +NA$
      |    |    └→ファイルディスクリプタ
      |    └→ファイル番号
      └→Input（インプット）：入力の意味です．
```

ファイル番号は，1～16まで選択できますが，1つのファイルに対するファイル番号は，全て，OPEN命令により，指定された番号を用います．

ファイルディスクリプタのファイル名は，行番号160のINPUT命令により，NA$という文字変数へ記憶したものを用います．

たとえば，行番号160において，"DATA"と入力したとすれば，ファイルディスクリプタは，"CASO：DATA"となります．

190　CONSOLE　6，0

画面をCONSOLE命令により，上下2つに分割しています．

```
CONSOLE 6, 0
        |  └→スクロール行数
        └→スクロール開始行
```

ここでは，6行目より，スクロール行数を0と設定していますので，2つのページモードの画面となっています．

200　LOCATE　0，0：PRINT "ファイルメイ……" +NA$

画面の左上へファイル名を表示させます．

210 LOCATE 8, 2 : PRINT "テンメイ" SPC(12) "デンワバンゴウ"

画面の2行目へ店名と電話番号という項目名を表示します．

220 INPUT＃1, A$, B$

カセットテープのデータファイルより，A$，B$という文字変数へデータを入力します．

```
INPUT#1, A$, B$
      ──  ──────
      │   └→変数名
      └→ファイル番号
```

230 LOCATE 6, 3 : PRINT USING "&　　　&　　&　　　&" ; A$, B$

画面の3行目へ，行番号220のINPUT＃命令により，A$とB$へ入力されたファイルのデータをPRINT USING命令により表示します．

```
                          ┌→この間は空白が出力されます
                       ┌──┴──┐
PRINT USING "&       &    &       & "; A$, B$
             ─────────    ─────────   ───────
             │                 │       └→出力する文字変数
             ├─────────────────┘
             └→&(ampersand：アンパサント)
               記号により囲まれた空白の数に2を
               加えた数の文字を，文字変数A$と
               B$の内容より表示します．
```

240 LOCATE 8, 5 : PRINT "ヒンメイ" SPC(12) "カカク";

画面の5行目へ，品名と価格という項目名を表示します．" ;"(semicolon：セミコロン)は，改行を行なわないことを指定します．

250 LOCATE 0, 6

画面の6行目左端へカーソルの位置を指定します．今までの画面表示は，CONSOLE命令により上下2つに分割した画面のうち，上の画面へ表示を行なっていましたが，このLOCATE命令により，以後画面出力は，下の画面であることを指定します．

260 INPUT＃1, A$, B$

カセットテープのデータファイルより，A$，B$という文字変数へデータを入力します．

```
INPUT #1, A$, B$
      ──  ──────
      │   └→変数名
      └→ファイル番号
```

270 PRINT SPC(6); : PRINT USING "&　　　&　　&　　　&" ; A$, B$

PRINT命令により空白を6個分出力します，" ;"（セミコロン）が記述されておりますので，次のPRINT USING命令の出力は同じ行に出力されます．

PRINT USING命令は，行番号260のINPUT＃命令により，ファイルのデータを記憶してある文字変数A$，B$の内容を表示します．

280　IF　EOF(1)　THEN　GOTO　290　ELSE　260

EOF関数は，シーケンシャルファイルよりデータを入力する際，入力したデータがそのファイルの最後であるかの検出を行ないます．

最後であれば −1(真)を，そうでなければ 0(偽)を与えます．

この行では，EOF関数により，読んだデータが，ファイルの最後のデータであれば，行番号290へ，最後でなければ行番号260へ実行を移します．

290　PRINT "データ　ヲ　スベテ　ヨミオワリマシタ"

行番号280のIF文により，ファイルのデータを全て読むと，この行へ実行が移り，PRINT命令によりファイルを全て読み終ったことのメッセージを出力します．

300　CLOSE

ファイルをCLOSE命令により閉じます．

310 PRINT "ガメン　ヲ　ケスナラ　Y"

画面へデータを表示したままにするかのメッセージを出力します．

320　IF　INKEY$　<>　"Y"　THEN　320

[Y] が押されるまでこの行で止まっています．この行により，画面へデータを表示したままとなっています．

330　WIDTH　40, 20

行番号320のIF文により，[Y] を押すと，この行番号へ実行が移り，WIDTH命令により画面をクリアし，キャラクタ表示を横40桁，縦20行に設定します．

この命令では，行番号190のCONSOLE命令により設定された画面表示の解除を行ないます．

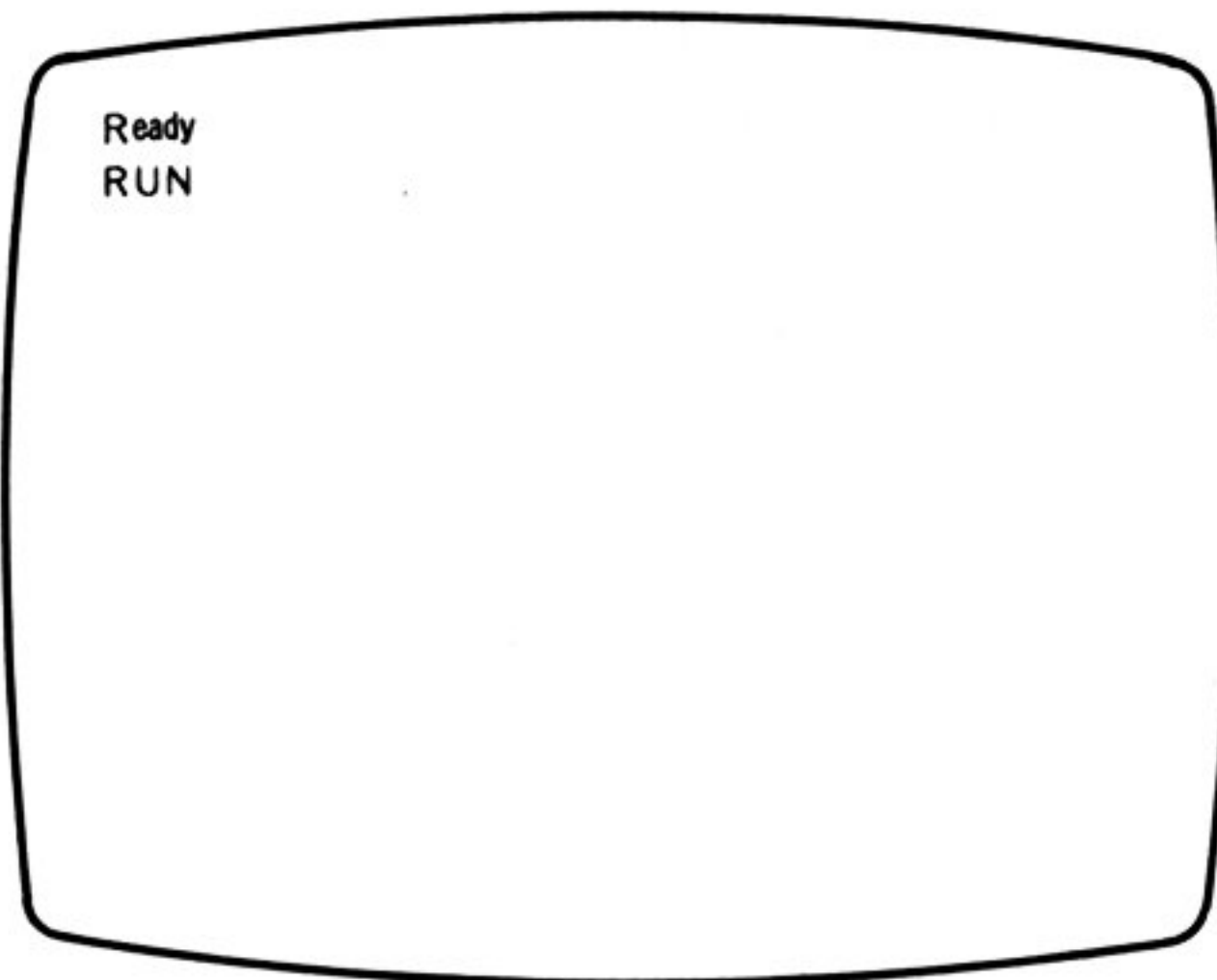
340　END

実行の終了を指定します．

例に示したプログラムの実行例

プログラムを実行します．

RUN [↵]

```
Ready
RUN
```

画面へカセットテープを準備するように，メッセージが出力されます．このメッセージにより，カセットテープを巻き戻し，PLAYボタンを押します．

カセットテープは，リモート機能により，停止しています．

準備完了であれば，[Y] を押します．

```
カセット テープ ハ OK デスカ?
OK ナラ Y
```

ファイル名を入力するように，メッセージを出力します．

[例]

エフエムヤ[↵]と入力します．

ファイル名の後には，必ず[↵]を入力します．

```
ファイルメイ ヲ ニュウリョクシテクダサイ
```

ファイル名を入力後，メッセージを出力して，カセットテープが回り始め，ファイルを探します．

```
イマ ファイル ヲ サガシテイマス
```

ファイルを見つけて，バッファが一杯(255バイト）になるまでデータを読むと，カセットテープは停止します．

カセットテープの停止後，画面へファイルのデータを出力します．

画面へ品名を14行分出力すると，CONSOLE命令により画面がページモードになっておりますので，7行目以降のデータを消し，新しく7行目へデータを出力します．

画面出力の一時停止を行なう場合は，[ESC]を押して下さい．

バッファ内のデータを全て出力すると，カセットテープが回り始め，次のデータを読みます．

最後に，データを全て読み終ったことのメッセージと，画面を消すかどうかのメッセージを出力します．

画面を消すなら[Y]を押します．

```
ファイルメイ－－－－－－－－エフエムヤ

        テンメイ                デンワバンゴウ
        エフエムヤ              00-111-2222

        ヒンメイ                カカク
        テンプラテイショク      1200
        トロロテイショク        1000
        ソバテイショク          600
        ソバベントウ            800
        テンザラ                1000
```

```
ファイルメイ－－－－－－－－エフエムヤ

        テンメイ                デンワバンゴウ
        エフエムヤ              00-111-2222

        ヒンメイ                カカク
        オヤコドンジョウ        900
        オヤコドンナミ          600
        テントジドン            600
        カイカドン              600
        カレーライス            500
        タマゴドン              500
        タヌキドン              500
        キツネドン              500
        カレードン              500
        ナメコジル              250

データ ヲ スベテ ヨミオワリマシタ
ガメン ヲ ケスナラ Y
```

```
Ready
```

12.7 MERGE 命令と CHAIN 命令の使い方

12.7.1 MERGE（マージ）

F-BASICにより作成された，メモリ上のプログラムと，すでにカセットテープ上へアスキー形式によりセーブを行なったプログラムファイルとの結合を行なう命令がMERGE（マージ）命令です．また，プログラムの結合を行なうことを〝マージする〟といいます．

MERGE 〝ファイルディスクリプタ〟, R

- 〝ファイルディスクリプタ〟：マージを行なおうとする，カセットテープ上のアスキー形式プログラムファイルを指定します
- R：マージ終了後，マージされたプログラムの実行を指定します

MERGE命令を使用したプログラム例を示し，MERGE命令に関する説明を行ないます．

プログラム例

マージするプログラム

```
20 CLS
30 A$="Blue"
40 B$="Bird"
50 PRINT A$;" ";B$
60 PRINT"ｶｾｯﾄﾃｰﾌﾟ OK?"
70 PRINT"OK ﾅﾗ Y"
80 IF INKEY$<>"Y" THEN GOTO 80
90 MERGE"CAS0:REA",R
```

マージされるプログラム
ファイル名〝REA〟

```
10 GOTO 90
90 PRINT"ｼｱﾜｾ ｦ ﾖﾌﾞ"
100 PRINT"ｱｵｲﾄﾘ"
110 PRINT A$;" ";B$
```

プログラム説明

20 CLS

画面をクリアします．

30 A$=〝Blue〟

40 B$=〝Bird〟

文字変数A$とB$へ文字データを与えます．

50 PRINT A$; 〝 〟; B$

画面へデータを出力します．

60 PRINT 〝カセットテープOK？〟

70 PRINT 〝OKナラY〟

80 IF INKE$ ◇〝 Y〟 THEN GOTO 80

10 GOTO 90

行番号90へ実行を移します．

90 PRINT 〝シアワセ ヲ ヨブ〟

100 PRINT 〝アオイトリ〟

110 PRINT A$; 〝 〟; B$

画面へ文字データとA$，B$の内容を出力します．

カセットテープを準備するようメッセージを出力します.

カセットテープを使用する際，必ずこのようなメッセージと，準備できるまでプログラムの実行を停止させておく処理が必要です.

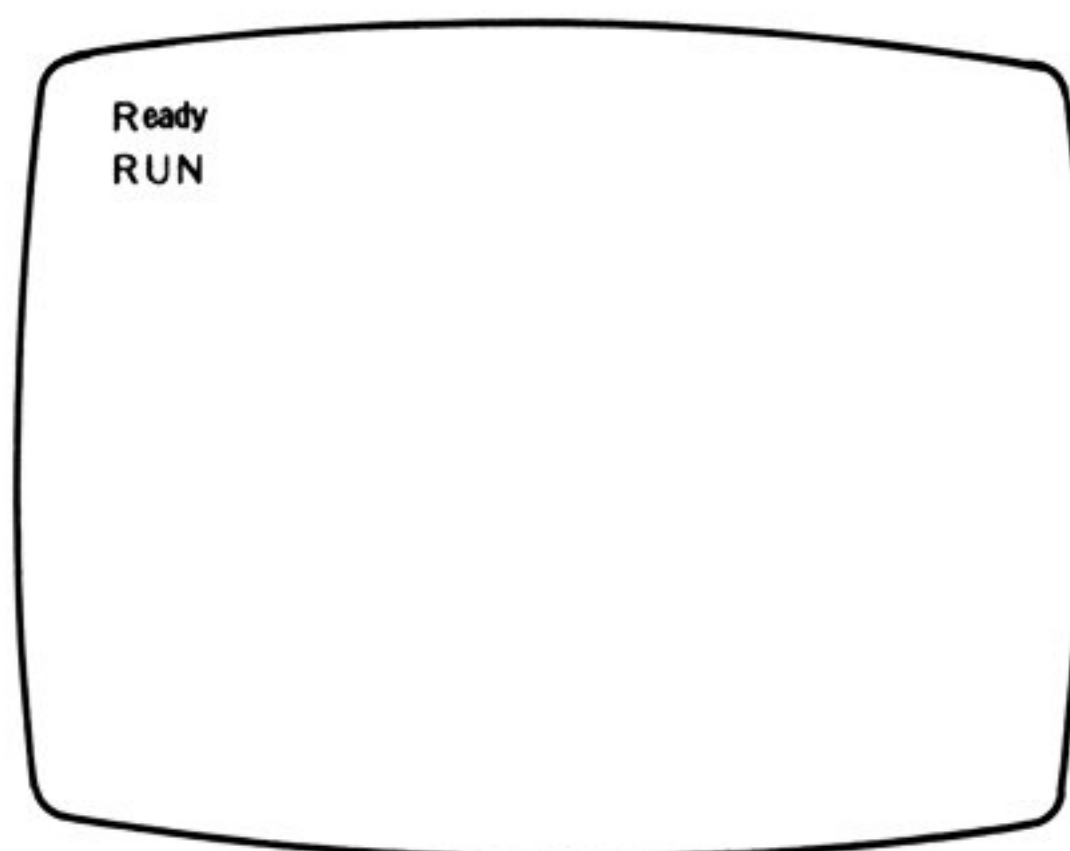
90 MERGE "CASO：REA", R

ファイル名 "REA" をマージし，マージ終了後プログラムの実行を開始します.

このプログラムの実行例を示します.

プログラムを実行します.

RUN ⏎

```
Ready
RUN
```

画面が消え，1行目に "Blue Bird", 2行目と3行目にメッセージが出力されます.

1行目の表示は，行番号30～50によるものです.

カセットテープの準備完了であれば，Y を押します.

```
Blue  Bird
カセットテープ OK?
OK ナラ Y
```

カセットテープが回り始め，行番号 90 の MERGE 命令を実行します．

マージ終了後，プログラムを実行します．それにより，文字データが出力されます．

しかし，行番号 110 のデータは何も出力されていません．MERGE 命令におけるマージ後の実行指定は，変数を受け渡さないのです．

つまり，マージされた後のプログラムを，新たに実行することになります．

マージ後のリストを示します．このリストでは，行番号 10 から行番号 90 へ実行が移り，文字変数 A$，B$ へデータは何も定義付けられません．

その A$，B$ を行番号 110 において PRINT 命令により表示させても，何も出力されないのです．

つまり MERGE 命令は，プログラムの結合のみを行ないます．

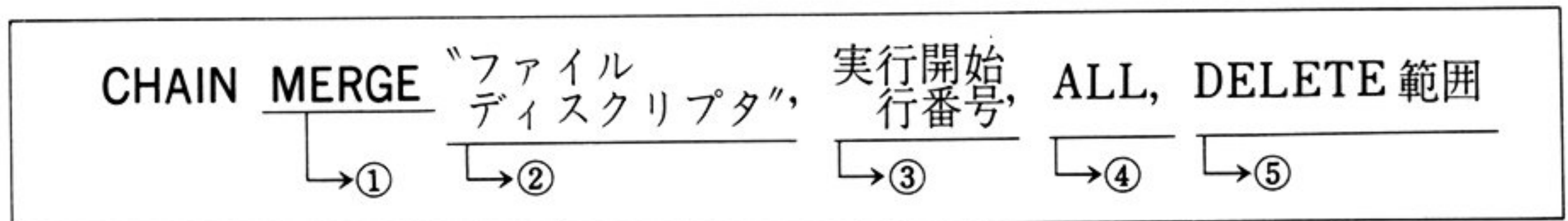
```
Blue  Bird
ｶｾｯﾄﾃｰﾌﾟ OK?
OK ﾅﾗ Y
ｼｱﾜｾ ｦ ﾖﾌﾞ
ｱｵｲﾄﾘ


Ready
```

```
10 GOTO 90
20 CLS
30 A$="Blue"
40 B$="Bird"
50 PRINT A$;" ";B$
60 PRINT"ｶｾｯﾄﾃｰﾌﾟ OK?"
70 PRINT"OK ﾅﾗ Y"
80 IF INKEY$<>"Y" THEN GOTO 80
90 PRINT"ｼｱﾜｾ ｦ ﾖﾌﾞ"
100 PRINT"ｱｵｲﾄﾘ"
110 PRINT A$;" ";B$
```

12.7.2 CHAIN（チェイン）・COMMON（コモン）

複数のプログラムを次々と連結させ，1つのプログラムのように実行させる命令が，CHAIN(チェイン）命令です．プログラムを連結させることを，〝チェインする〟といいます．

CHAIN 命令には，プログラムをマージさせる機能と，変数を渡す機能，チェインしたプログラムを任意の行番号より実行させる機能および，任意の行番号を消去する機能を持っています．

CHAIN MERGE "ファイルディスクリプタ", 実行開始行番号, ALL, DELETE 範囲
（MERGE→①　"ファイルディスクリプタ"→②　実行開始行番号→③　ALL→④　DELETE 範囲→⑤）

① MERGE（マージ）

メモリ上のプログラムと，ファイルディスクリプタにより指定したプログラムファイルとをマージする場合に指定します．

前述の MERGE 命令とマージする方法は同様であり，マージするプログラムは，アスキー形式でなければなりません．

ただし，バイナリ形式のプログラムファイルをマージする場合，そのプログラムファイルの先頭の行番号は，メモリ上のプログラムの最大行番号より大きくなければなりません．

② ファイルディスクリプタ

カセットテープ上のプログラムファイルを指定します．ROMモードであれば，デバイス名は省略できます．

③ 実行開始行番号

チェインした後のプログラムを実行する行番号を指定します．省略した場合は，最初の行番号より実行を開始します．

④ ALL（オール）

変数を全てチェインした後のプログラムへ渡す場合，〝ALL〞と記述します．省略した場合，渡す変数は，COMMON文により，指定しなければなりません．COMMON文も省略すると，変数は渡しません．

⑤ DELETE（デリート）範囲

チェインを行なう際，メモリ上のプログラムにおける不要な行番号を消去するための指定を行ないます．

チェインは，指定した行番号の消去を行なった後，実行します．

COMMON 変数名

COMMON文は，CHAIN命令と共に利用され，CHAIN命令が実行された時，指定した変数をチェインするプログラムへ渡します．

COMMON文の記述例

```
COMMON A, B, C$, D$, E( ), F$( )
```

変数名は〝，〞（コンマ：comma）により区切り記述します．また，配列変数には，必ず〝(　)〞を与えます．

CHAIN, COMMON文の記述例

メモリ上のプログラム

```
10 DIM A(10)
20 B$="ABCDE"
30 C= 12345
40 FOR I=1 TO 10
50 A(I)=I*100
60 NEXT
70 PRINT"ｶｾｯﾄﾃｰﾌﾟ OK?"
80 PRINT"OK ﾅﾗ Y"
90 S$=INKEY$
100 IF S$<>"Y" THEN 90
110 COMMON A(),B$,C
120 CHAIN"CASO:SUB",10
130 END
```

行番号10～60

配列変数A( ), 文字変数B$数値変数Cへデータを与えます.

行番号70～100

カセットテープを使用する際のメッセージと, 準備できるまで待つような処理を行ないます.

行番号 110

COMMON文により, 渡す変数であるA( ), B$, Cを指定します.

行番号 120

チェインするプログラムファイルとチェイン後のプログラムを実行する行番号を指定します.

行番号 130

プログラムの終了を示します.

チェインされるプログラム

```
10 PRINT
20 FOR I=1 TO 10
30 PRINT"A(";STR$(I);")";"=";
40 PRINT A(I)SPC(2);
50 NEXT
60 PRINT:PRINT
70 PRINT"B$=";B$:PRINT
80 PRINT"C=";C
90 END
```

行番号 10, 60

改行を行ないます.

行番号 20～50

配列変数A( )のデータを表示します.

行番号 70

文字変数B$のデータを表示します.

行番号 80

数値変数Cのデータを表示します.

行番号 90

プログラムの終了を示します.

行番号110, および120において, COMMON文により全てのデータを渡すように指定を行なっておりますので, これを, CHAIN命令のALL指定を行なうことにより, COMMON文を省略することができます.

```
110 COMMON A( ), B$, C      }
120 CHAIN "CASO : SUB", 10  } ⇨ 110 CHAIN "CASO : SUB", , ALL
```

実行開始行番号の10は, チェイン後のプログラムにおける実行開始行番号ですので省略できます.

MERGEおよびDELETEの指定

MERGEの指定により, チェインするプログラムをマージできます. また, DELETEの指定により, 行番号の消去を行ないます.

以下に, MERGEとDELETEの指定を持つCHAIN命令を使用したプログラムの記述例を示します.

[例1] アスキー形式ファイルのチェイン

メモリ上のプログラム

アスキー形式のプログラムファイルをチェインします.

```
10 DATA 36,24,68,99,7,18,8,41,53,87
20 KOSU$="ｺｽｳ ﾊ    "
30 MDAT$="ﾃﾞｰﾀ ﾊ   "
40 SGOK$="ｺﾞｳｹｲ ﾊ "
50 HEIK$="ﾍｲｷﾝ ﾊ   "
60 DIM DDAT(10)
70 HEIKI=0
80 SGOKE=0
90 KOSUU=10
100 FOR I=1 TO KOSUU
110 READ DDAT(I)
120 NEXT
200 PRINT KOSU$;KOSUU
210 PRINT MDAT$
220 FOR I=1 TO KOSUU
230 PRINT DDAT(I);
240 NEXT
250 PRINT
260 COMMON KOSU$,KOSUU,MDAT$,DDAT()
270 COMMON SGOK$,HEIK$
280 PRINT"ｶｾｯﾄ ﾊ OK?"
290 PRINT"OK ﾅﾗ Y"
300 IF INKEY$<>"Y" THEN GOTO 300
310 CHAIN "CAS0:SUB1"
320 END
```

チェインするプログラム（ファイル名：SUB 1）

```
200 '*** SUB1 ***
210 FOR I=1 TO KOSUU
220 SGOKE=SGOKE+DDAT(I)
230 NEXT
240 HEIKIN=SGOKE/KOSUU
250 PRINT SGOK$;SGOKE
260 PRINT HEIK$;HEIKIN
270 END
```

[例2] アスキー形式ファイルのチェイン

メモリ上のプログラム

アスキー形式のプログラムファイルを行番号200〜300の間へチェインします.

```
10 DIM DDATA(100)
20 DIM DDAT$(100)
30 KOSU$="ｺｽｳ ﾊ  "
40 MDAT$="ﾃﾞｰﾀ ﾊ "
50 SGOK$="ｺﾞｳｹｲ ﾊ"
60 HEIK$="ﾍｲｷﾝ ﾊ "
70 HEIKI=0
80 SGOKE=0
90 PRINT"ﾃﾞｰﾀｦ ﾆｭｳﾘｮｸ ｼﾃｸﾀﾞｻｲ"
100 PRINT"ﾆｭｳﾘｮｸﾉ ｵﾜﾘﾊ 'E' ﾃﾞｽ"
110 KOSUU=1
120 INPUT "DATA=";IDAT$
130 IF IDAT$="E" THEN GOTO 170
140 DDATA(KOSUU)=VAL(IDAT$)
150 KOSUU=KOSUU+1
160 GO TO 120
170 KOSUU=KOSUU-1
290 PRINT"ｶｾｯﾄﾃｰﾌﾟ ﾊ OK?"
295 PRINT"OK ﾅﾗ Y"
296 IF INKEY$<>"Y" THEN GOTO 296
300 CHAIN MERGE"CAS0:SUB2",200,ALL
310 PRINT
320 PRINT"ﾂﾂﾞｹﾏｽｶ? Y/N"
330 F$=INPUT$(1)
340 IF F$="Y" THEN GOTO 70
350 END
```

MERGE
ファイル名 "SUB2" のプログラムファイルをマージします.
実行開始行番号
チェイン後のプログラムの行番号 200 より実行します.
ALL
全ての変数をチェインするプログラムへ渡します.

最初にチェインするプログラム

```
200 '*** SUB2 ***
210 PRINT:PRINT KOSU$;KOSUU
220 PRINT:PRINT MDAT$
230 FOR I=1 TO KOSUU
240 PRINT DDATA(I);
250 NEXT
260 PRINT
300 CHAIN MERGE"1:SUB3",220,ALL,DELETE 200-270
```

MERGE
ファイル名 "SUB3" のプログラムファイルをマージします.
実行開始行番号
チェイン後のプログラムの行番号 220 より実行します.
ALL
全ての変数をチェインするプログラムへ渡します.
DELETE 範囲
メモリ上のプログラムの行番号 200～270 までを消去します.

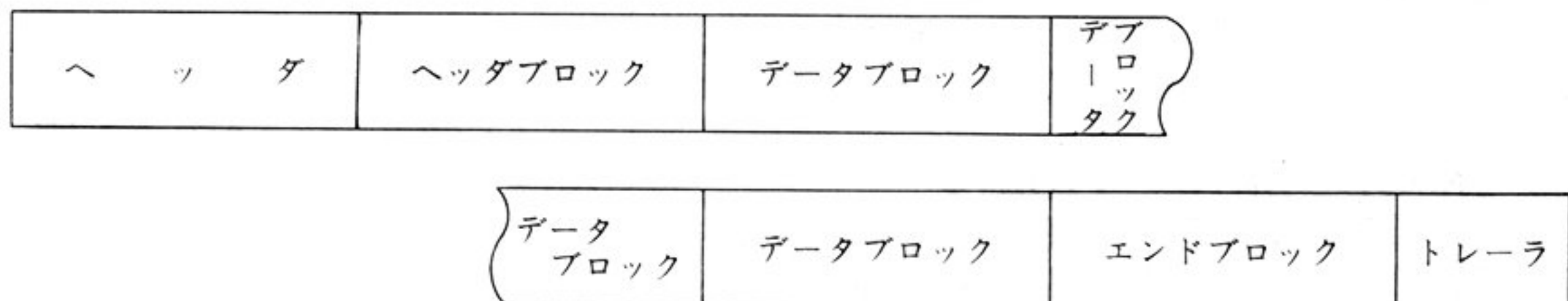
2番目にチェインするプログラム

```
200 '*** SUB3 ***
210 GOTO 290
220 FOR I=1 TO KOSUU
230 SGOKE=SGOKE+DDATA(I)
240 NEXT
250 HEIKIN=SGOKE/KOSUU
260 PRINT:PRINT SGOK$;SGOKE
270 PRINT:PRINT HEIK$;HEIKIN
280 GO TO 310
300 CHAIN MERGE"1:SUB2",200,ALL,DELETE200-280
```

12.8 カセットテープにおけるデータの記録形式

F-BASICにおいては，データを，カセットテープに対して，以下に示す形式により記録します．

ファイルフォーマット（ファイルの記録形式）

ヘッダ	ヘッダブロック	データブロック	データブロック	データブロック	データブロック	エンドブロック	トレーラ

・ヘッダ

カセットテープのクリアフィーダ（磁性体の塗られてない所）を送るためと，前のファイルとの間隔を開けるために無録音状態により，カセットテープを送ります．

ヘッダは，約10秒の無録音状態です．

・ヘッダブロック

ファイル名，ファイルの記録形式（アスキーまたは，バイナリ形式）が記録されます．

・データブロック

データが記録されます．

・エンドブロック

ファイルの終了を示します．

・トレーラ

ヘッダと同様な無録音状態となっており，エンドブロックが，次のファイルを記録した場合，消去されないように，余裕を持たせたものです．

トレーラは，約1秒の無録音状態です．

ブロックフォーマット（各ブロックの記録形式）

ギャップ	0 1	3 C	ブロックタイプ	ブロック長	データ	チェックサム	ギャップ

各ブロックには，前後にギャップがあり，ここにはFF(16進数）が書かれています．先頭のギャップの次には，013C(16進数）と書かれています．F-BASICはこの一連の記録形式である，FF013C(16進数）をブロックの始まりだと判断します．

・ブロックタイプ

ブロックの種類	記録される16進データ
ヘッダブロック	0 0
データブロック	0 1
エンドブロック	F F

・ブロック長

データの長さ（バイト数）を1バイトの16進数により記録します(0 0～F F：16進数)

・データ

255バイトまでのデータを記録します．

ヘッダブロックの場合は，ファイル名，ファイルの記録形式（アスキーまたはバイナリ形式）が記録されます．

エンドブロックの場合は，データを何も記録しません．

・チェックサム

ブロックタイプ，ブロック長，データを全て8ビットの範囲において加算します．8ビットを越えたものは無視し，加算し終った8ビットをチェックサムのデータとして記録します．

チェックサムは，スタートビットが1ビット，ストップビットが2ビット付加され，11ビットにより構成されます．

0	データ　8ビット	1	1

スタートビット　　ストップビット

第13章　ミニフロッピィディスク

13.1　ファイルディスクリプタ

ミニフロッピィディスクに対して，プログラムのロード・セーブや，データの入出力を行なう際，ファイルディスクリプタを指定します。

ファイルディスクリプタは，デバイス名とファイル名により構成されます．

また，ミニフロッピィディスクユニットのドライブを指定する場合があります．ドライブは，ドライブ番号により指定し，ドライブ番号は，各ドライブに対して決まっています．

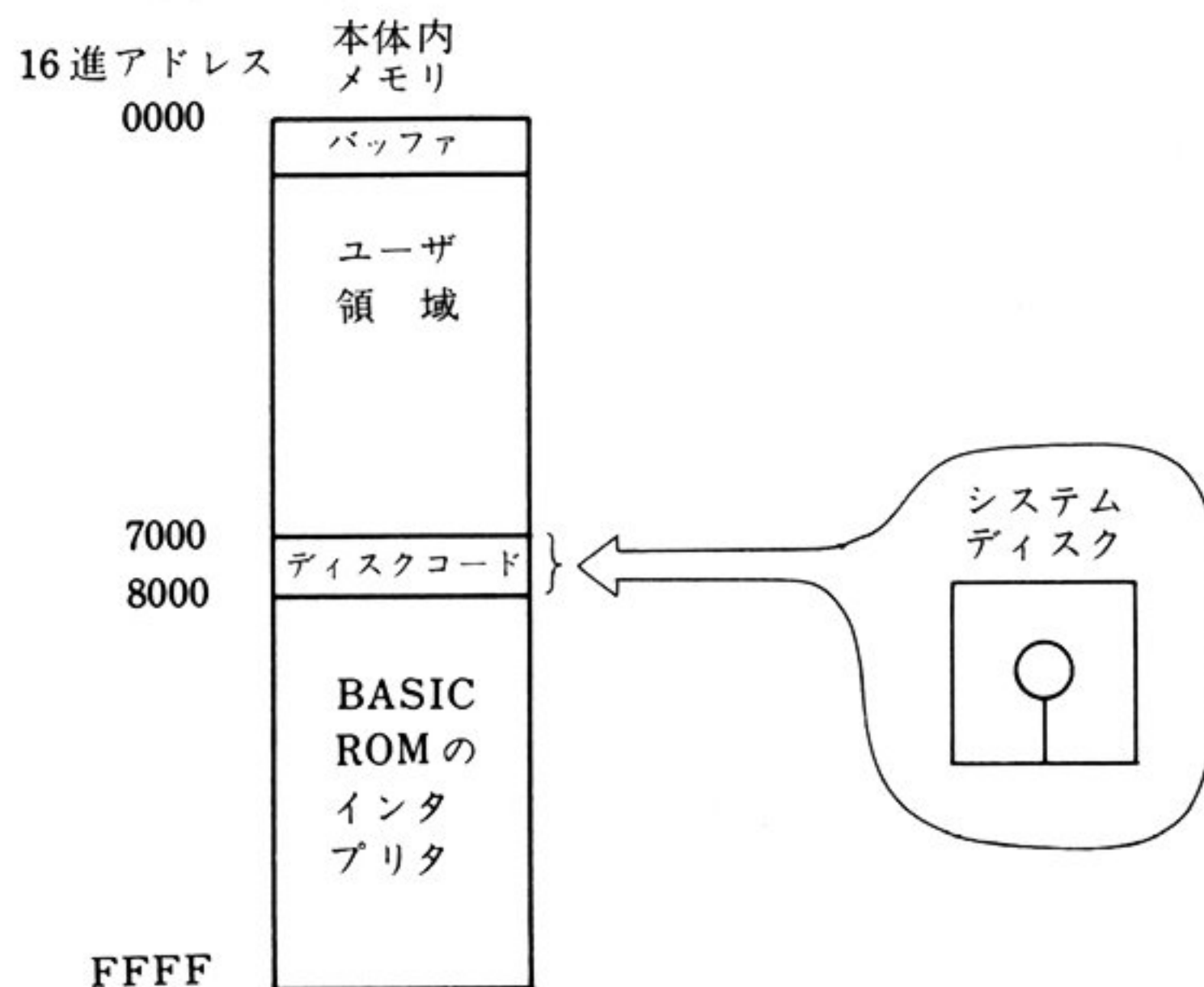
以下に，ドライブに対するデバイス名，およびドライブ番号を示します．

デバイス名

ドライブ名称	デバイス名	ドライブ番号
ドライブ0	0：	0
ドライブ1	1：	1

ミニフロッピィディスクを使用する場合は，必ず，システムディスクより，F-BASIC V3.0 DISK モード（以下 DISK モードと記します．）の起動を行ないます．

これは，DISK モードを起動する際，ミニフロッピィディスクに対する，データ入出力用のバッファ

16進アドレス　本体内メモリ
0000　バッファ
ユーザ領域
7000　ディスクコード
8000
BASIC ROMのインタプリタ
FFFF
システムディスク

を設定したり，システムディスクより，ミニフロッピィディスク専用の命令を使用可能とするためのプログラム（ディスクコード）をメモリ上へ展開するといった，ミニフロッピィディスクを使用するための準備を行なうからです．

DISKモードのF-BASICは，BASIC ROMに書かれたプログラム（インタプリタ）と，システムディスクよりメモリ上に展開されたディスクコードの両方を使用します．

13.2 ファイル名の出力方法 (FILES)

ミニフロッピィディスクに記録されているファイルのファイル名，ファイル形式，ファイルの種類，ファイルの記録形式と，さらにファイルのミニフロッピィディスクに対する専有量を出力する命令が〝FILES（ファイルズ）″です．

```
FILES "    "
       └── デバイス名を指定します.
```

ディスプレイへの出力例

```
Ready
FILES"0:"

DEMO        0 B S 2

150 Clusters Free

Ready
█
```

●出力された″DEMO 0BS2″の意味

```
DEMO 0  B  S  2
```

- 2 ── 専有量　ファイルの使用クラスタ数を表します．クラスタとは，F-BASICがミニフロッピィディスクにおけるファイルの専有量を管理する単位です．1クラスタは1/2トラックで，2048バイト（2Kバイト）となります．〝DEMO″プログラムは，2クラスタ分専有しております．
- S ── ファイル形式　S：シーケンシャルファイル　R：ランダムファイル
- B ── ファイルの記録形式　B：バイナリ形式　A：アスキー形式
- 0 ── ファイルの種類　0：BASICプログラムファイル　1：データファイル　2：機械語プログラムファイル
- DEMO ── ファイル名

●150 Clusters Freeの意味

ミニフロッピィディスクの残りのクラスタ数を表わします．残りは，150クラスタあり，1クラスタは2Kバイトですので，2(Kバイト)×150(クラスタ)＝300Kバイトのデータをこのミニフロッピィディスクへ記録することができます．

ファイルが，まったく記録されていないミニフロッピィディスクは152クラスタ持っております．F-BASICは，クラスタ単位により，ファイルの管理を行ないますので，ファイルを記録できる最大数は，152となります．

13.3 ファイル名の変更(NAME)

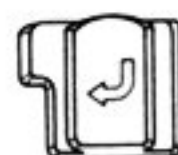
ミニフロッピィディスク上に記録されているファイルのファイル名を変更する場合使用する命令が〝NAME(ネーム)〟です．

```
NAME "________" AS "________"
       └─旧ファイル        └─新ファイル
         ディスクリプタ       ディスクリプタ
```

FILESによりファイル名を出力します．

```
Ready
FILES"0:"

DEMO       0 B S 1
DATA       1 A S 3

148 Clusters Free

Ready
■
```

上に表示された〝DEMO〟のファイル名を持つファイルを〝デモ〟というファイル名に変更する例を示します。

```
NAME "0:DEMO" AS "0:デモ"
```

デバイス名は，同一の指定を行ないます．

次に，変更後のファイル名を確認します．

```
Ready
FILES"0:"

ﾃﾞﾓ          0 B S 1
DATA        1 A S 3

148 Clusters Free

Ready
■
```

13.4 ファイルの消去

13.4.1 すべてのファイルの消去

ミニフロッピィディスクに記録されているファイルを，すべて消去する命令が "DSKINI（ディスクアイエヌアイ）" です．

```
DSKINI ______
         └───── ドライブ番号を指定します
```

ミニフロッピィディスクへ記録されているファイルを確認します．

```
Ready
FILES"0:"

ﾃﾞﾓ          0 B S 1
DATA        1 A S 3

148 Clusters Free

Ready
■
```

ファイルをすべて消去します．

```
DSKINI 0 [↵]
Are you sure (Y/N)?Y
```

Are you sure ── 消していいかの確認です

(Y/N)?Y ── 消していい時には "Y" を入力します

DSKINI でドライブ1のフロッピィディスクのファイルをすべて消しました. もう一度FILES でみてみましょう.

```
Ready
FILES"0:"

152 Clusters Free

Ready
■
```

以上のようにファイルは, すべて消えてしまいます.

13.4.2 指定したファイルの消去

ミニフロッピィディスクに対して, ファイルディスクリプタにより, 指定したファイルを消去する命令が "KILL (キル)" です.

```
KILL "____________"
        └── ファイルディスクリプタを指定します
```

以下に示すように, ドライブ0にDATA 100というデータファイルがあります. このデータファイルを消してみましょう.

```
Ready
FILES

DATA100    1 A S 3
DATA200    1 A S 10
DATA300    1 A S 5

134 Clusters Free

Ready
■
```

ファイル名 "DATA 100" のデータファイルを消去します.

```
KILL "0:DATA100" [↵]
      └── ファイルディスクリプタ＝デバイス名＋ファイル名
Aer you sure (Y/N)?
      └── DSKINI と同じように, 本当に消していいのかを確認します
```

ミニフロッピィディスクのファイルを見てみましょう.

```
Ready
FILES"0:"

DATA200    1 A S 10
DATA300    1 A S 5

137 Clusters Free

Ready
■
```

以上のようにKILLは, 指定したファイルのみの消去を行ないます.

13.5 プログラムファイルの扱い方

13.5.1 LOADとSAVE

F-BASICにより作成したプログラムを, ミニフロッピィディスクへ記録する命令が"SAVE(セーブ)"であり, 記録されているBASICプログラムファイルを本体メモリへ展開する命令が"LOAD(ロード)"です.

```
LOAD "        "
      └──ファイルディスクリプタの指定をします
```

ドライブ0のミニフロッピィディスクにある"DEMO"のファイル名を持つプログラムをロードします.

```
LOAD "0:DEMO" ↵
      └──ファイルディスクリプタ
```

LOADが実行された時, 以前メモリ上にあったプログラムはすべて消されてしまいます.

```
SAVE "        " [,A]
      │           └── ",A"と指定するとアスキー形式で記録します
      │               また, 指定がなければ, バイナリ形式で記録します
      └──ファイルディスクリプタを指定します
```

プログラムを，〝DEMO 1〞というファイル名によりセーブします．

```
SAVE "0:DEMO1"
```

ドライブ0のフロッピィディスクへメモリ上のプログラムを〝DEMO 1〞という名前で記録します．この時〝，A〞が指定されてないので，バイナリ形式により記録を行ないます．

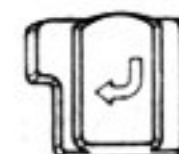
13.5.2 LOADMとSAVEM

〝LOAD〞と〝SAVE〞は，F-BASICにより作成されたプログラムファイルのロード・セーブを行ないましたが，機械語によるプログラムファイルのロード・セーブを行なう命令が〝LOADM（ロードエム）〞と〝SAVEM（セーブエム）〞です．

SAVEM〝　　〞，開始番地，終了番地，入口番地

- 〝　　〞：ファイルディスクリプタを指定します．
- 開始番地：記録を開始するメモリの番地を指定します．
- 終了番地：記録を終了するメモリの番地を指定します．
- 入口番地：プログラムの実行を開始する番地を指定します．

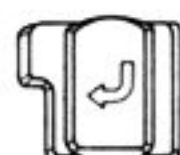
機械語によるプログラムが5000番地（16進）～5500番地（16進）までメモリ上に存在した場合，これを〝ML〞というファイル名でミニフロッピィディスクに記録する時の指定方法を示します．プログラムの入口番地は5000番地（16進）とします．

```
SAVEM "0:ML",&H5000,&H5500,&H5000
```

〝SAVEM〞コマンドは，機械語によるプログラムをミニフロッピィディスクに記録する際，開始番地，終了番地，入口番地も同時に記録します．

LOADM〝　　〞［,［オフセット値］［,R］］

- 〝　　〞：ファイルディスクリプタを指定します．
- オフセット値：SAVEMコマンドの開始番地に加えられる値です．
- R：メモリ上に書込み後入口番地よりプログラムの実行を行ないます．

指定した機械語によるプログラムファイルを，〝SAVEM〞で記録された開始番地から終了番地までメモリ上に書込みます．

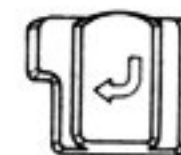
```
LOADM "1:ML"
```

MLと名付けられた機械語によるファイルを，〝SAVEM〞によって記録された開始番地から，終了番地までメモリ上に展開します．

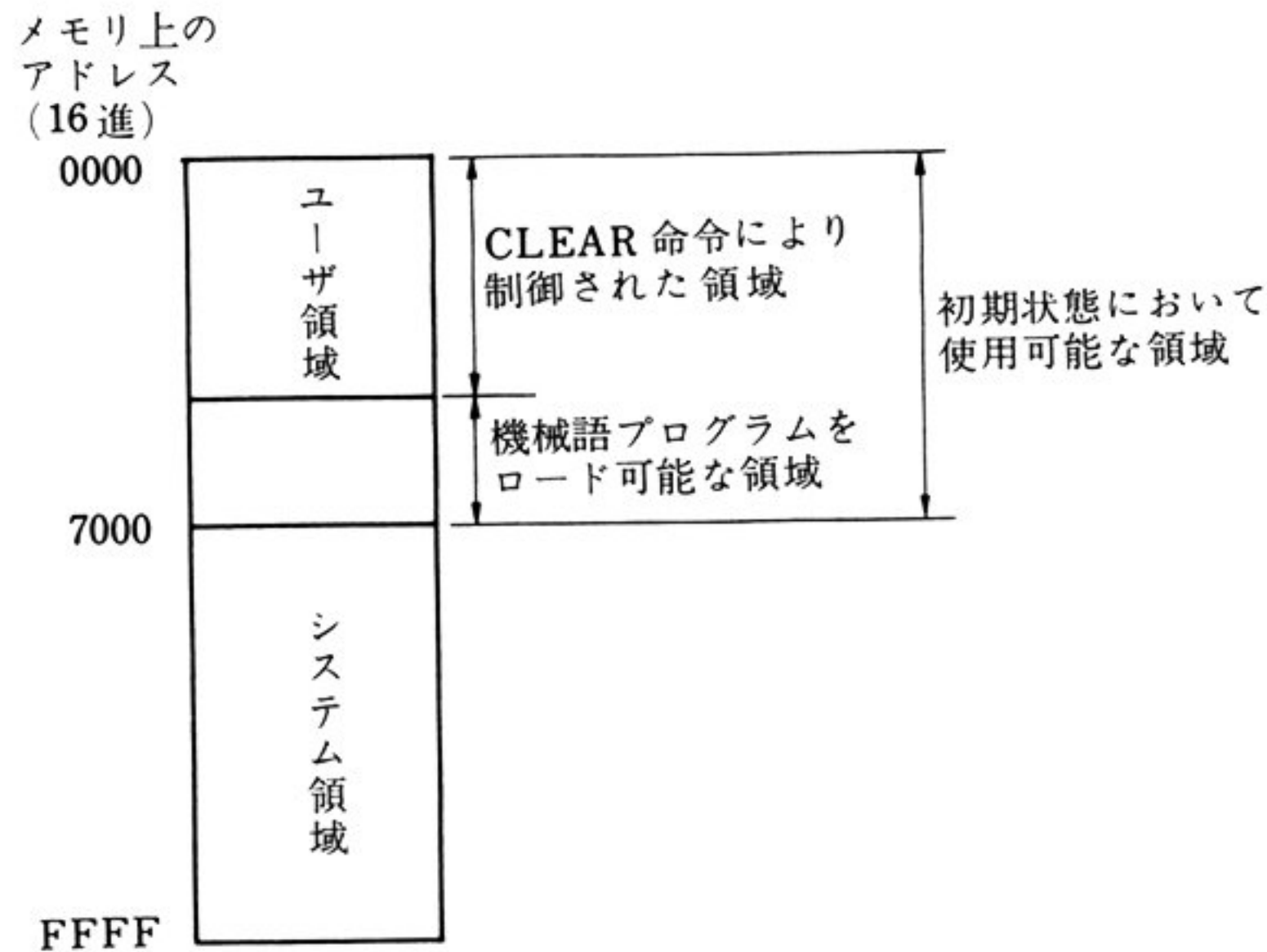
【SAVEM, LOADMの実行説明】

機械語プログラムが以下に示すメモリダンプリストのように6800番地(16進)よりメモリ上に存在した場合の実行例を説明します.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
6800	81	02	26	0E	EC	02	10	83	00	0D	27	6A	8D	05	8D	16	_.&.●..▬..'j█.█.
6810	8D	3F	39	ED	8D	00	7A	30	8D	00	78	AF	8D	00	70	30	█?9○█.z0█.xッ█.p0
6820	1A	AD	9F	FB	FA	39	30	8D	00	69	31	8D	00	89	8D	04	.ュ'..90█.i1█.|█.
6830	30	01	31	A5	5F	4F	68	86	66	A5	8B	02	81	10	26	F6	0.1·_0h■f·█._.&分
6840	5C	68	86	66	A5	8B	02	81	20	26	F6	5C	C1	10	26	E5	¥h■f·█._ &分¥チ.&◣
6850	39	CC	1B	34	ED	8D	00	5B	CC	00	10	ED	8D	00	56	30	97.4○█.[フ..○█.V0
6860	8D	00	50	AF	8D	00	0B	30	8D	00	05	AD	9F	FB	FA	39	█.Pッ█..0█..ュ'..9
6870	0E	00	00	00	00	24	30	8D	00	11	AF	8D	00	09	30	1A	.....$0█..ッ█..0.
6880	AD	9F	FB	FA	39	0E	00	00	00	00	02	0D	0A	16	00	00	ュ'..9...........
6890	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
68A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
68B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
68C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
68D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
68E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
68F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....

上に示す機械語プログラムを〝SAVEM〞により，6800番地(16進) 6900番地(16進)までフロッピィディスクへ格納します．入口番地は6800番地(16進)とします．

SAVEM "0：ML", &H6800, &H6900, &H6800 ⏎

セーブが終了した後，再び上記の格納したプログラムをメモリへロードします．ここで，機械語プログラムを，F-BASICにより作成したプログラムのサブルーチンとして使用する等，BASICプログラムと並用して使用する場合，機械語プログラムをメモリへロードする前に，〝CLEAR(クリア)〞命令により，F-BASICで使用するメモリの上限を設定し，機械語プログラムのメモリ上での領域を確保して下さい．

これは，F-BASICにおいて使用するメモリの領域と，機械語プログラムを展開するメモリの領域が，同一となった場合，メモリ上に存在していた，F-BASICのデータ，または機械語プログラムを破壊してしまう可能性があるからです．

CLEAR命令の指定方法を以下に示します.

```
CLEAR 300, &H67FF
```

上記のように設定すると, 6800番地(16進)以上の領域へ機械語プログラムをロードできます.

```
LOADM "0:ML"
```

機械語プログラムをメモリへロードします. この場合, "オフセット, R" の指定がないので, ロードし終われば, コマンド受付け状態へ戻ります.

13.6 データファイルの扱い方

ミニフロッピィディスクに対してのデータファイルを扱うために必要な処理を以下に示します.

(1) データをファイルへ記録(書く)する処理.

(2) データをファイルより入力(読む)する処理.

(3) ファイルのデータを修正する処理.

(4) ファイルへデータを追加する処理.

以上に示す4種類の処理のプログラミングを理解できれば, データファイルを有効に利用できます.

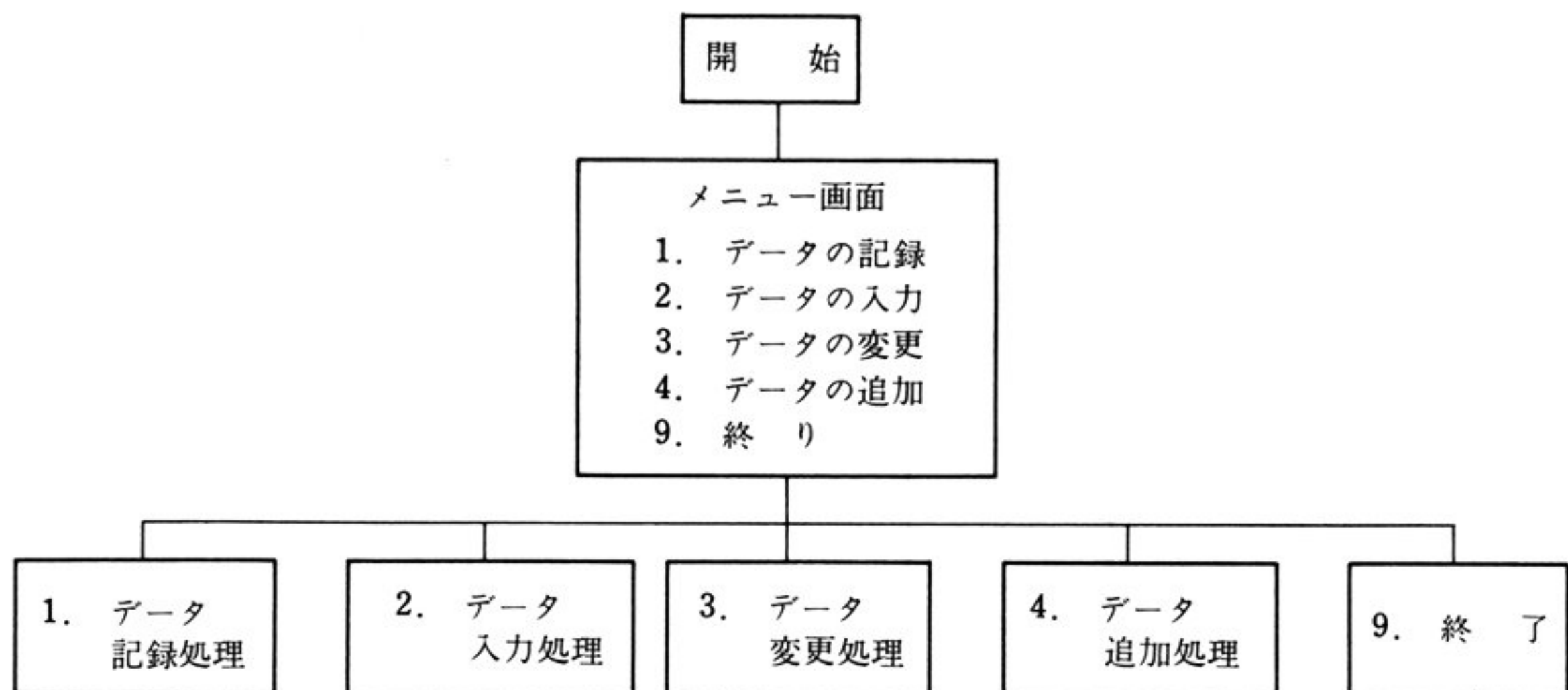
次に, データファイルを利用した, 住所録のプログラムを例として示し, シーケンシャルファイルとランダムファイルの説明を行ないます.

13.6.1 シーケンシャルファイルの扱い方

シーケンシャルファイルを利用したプログラムの処理内容を次に示します.

開　始

メニュー画面
1. データの記録
2. データの入力
3. データの変更
4. データの追加
9. 終　り

1. データ記録処理 | 2. データ入力処理 | 3. データ変更処理 | 4. データ追加処理 | 9. 終　了

プログラムは4つの処理に分かれています.

1. データ記録処理　　シーケンシャルファイルへデータを記録する.
2. データ入力処理　　シーケンシャルファイルよりデータを入力する.
3. データ変更処理　　シーケンシャルファイルよりデータを入力し変更する.
4. データ追加処理　　シーケンシャルファイルへデータを追加する.

(1) メニュー

プログラム例により説明します.

〈処理内容〉

この処理においては，ディスプレイへ，メッセージを出力して，目的の処理の選択を行ない，その処理へ実行を移します.

〈プログラム例〉

```
10 'ｼﾞｭｳｼｮﾛｸ ﾌﾟﾛｸﾞﾗﾑ
20 'ﾒﾆｭｰｶﾞﾒﾝ
30 '
40 WIDTH 40,20
50 SYMBOL(0,0),"♥♥♥ ｼﾞｭｳｼｮﾛｸ♥♥♥",4,2,4
60 LOCATE 4,4:PRINT" 1: ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ"
70 PRINT"       2: ﾃﾞｰﾀ ﾉ ﾖﾐﾄﾘ"
80 PRINT"       3: ﾃﾞｰﾀ ﾉ ﾍﾝｺｳ"
90 PRINT"       4: ﾃﾞｰﾀ ﾉ ﾂｲｶ"
100 PRINT"       9: ｵﾜﾘ"
110 INPUT"ｼｮﾘﾊﾞﾝｺﾞｳ ﾉ ﾆｭｳﾘｮｸ     ",Q
120 ON Q GOTO 160,390,540,830
130 INPUT"ｼｮﾘ ｦ ｵﾜﾘ ﾏｽｶ? Y/N     ",Q$
140 IF Q$<>"Y" THEN GOTO 150
144 GOSUB 370
145 LOCATE 0,12:PRINT"♠♠♠ ｵﾜﾘ ♠♠♠"
```

```
146 END
150 GOTO 40
```

〈プログラムの解説〉

・40～100 行　　メッセージを出力しています.

・110 行　　目的の処理を選択します.

・120 行　　目的の処理へ実行を移します.

・130～150 行　　プログラムを終了させるかの判定を行ないます.

〈処理上の画面表示〉

```
♥♥♥ ジュウショロク♥♥♥

        1: データ ノ トウロク
        2: データ ノ ヨミトリ
        3: データ ノ ヘンコウ
        4: データ ノ ツイカ
        9: オワリ
ショリバンゴウ ノ ニュウリョク      1
```

メニュー画面

(2) シーケンシャルファイルにデータを記録する（データ記録処理）

シーケンシャルファイルにデータを記録するためには，以下の命令が必要です.

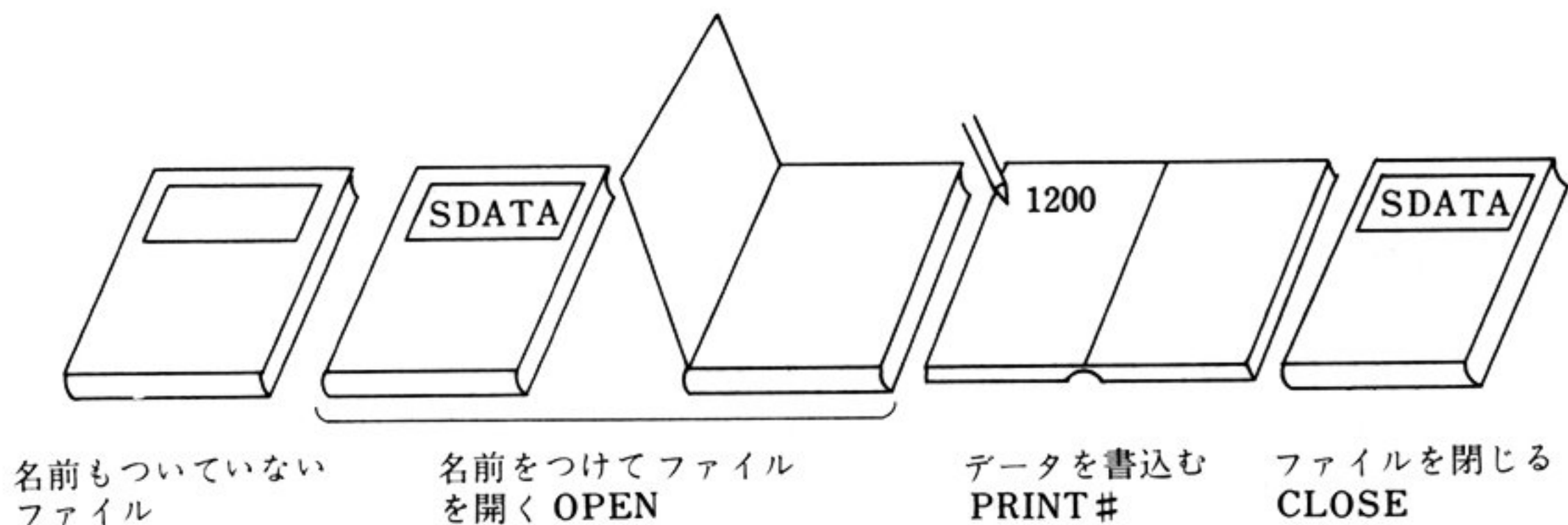
・書込むファイルに名前をつけファイルを開く OPEN 命令

・シーケンシャルファイルにデータを記録する PRINT #命令

・書込むデータがなくなった時ファイルを閉じる CLOSE 命令

この OPEN，PRINT #，CLOSE の 3 つの命令により構成されます.

SDATA　　1200　　SDATA

名前もついていないファイル　　名前をつけてファイルを開く OPEN　　データを書込む PRINT#　　ファイルを閉じる CLOSE

〈処理内容〉

登録するファイル名を入力し，シーケンシャルファイルをオープンします．記録するデータを入力

しまちがいがなければ，ファイルにデータを書込みます．入力するデータがなくなった時ファイルをクローズしてメニュー画面にもどります．

〈プログラム例〉

```
160 ' ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ
170 WIDTH 40,20
180 NO=1:PRINT"♣♣♣ ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ ♣♣♣"
190 LOCATE 0,18
195 INPUT"ﾄｳﾛｸｽﾙ ﾌｧｲﾙﾒｲ ﾉ ﾆｭｳﾘｮｸ   ",F$
200 OPEN"O",#1,"O:"+F$
210 GOSUB 370:LOCATE 0,18
220 INPUT"ｼﾒｲ ﾉ ﾆｭｳﾘｮｸ   ",NAM$
225 GOSUB 370:LOCATE 0,18
230 INPUT"ｼﾞｭｳｼｮ ﾉ ﾆｭｳﾘｮｸ   ",JYU$
235 GOSUB 370:LOCATE 0,18
240 INPUT"ﾃﾞﾝﾜﾊﾞﾝｺﾞｳ ﾉ ﾆｭｳﾘｮｸ   ",TEL$
245 GOSUB 380:LOCATE 2,5
250 PRINT NO
260 LOCATE 4,6:PRINT NAM$
270 LOCATE 6,7:PRINT JYU$
280 LOCATE 8,8:PRINT TEL$
290 LOCATE 0,18
295 INPUT"ﾃﾞｰﾀﾉｱﾔﾏﾘ ﾊ ﾅｲﾃﾞｽｶ? Y/N  ",Q$
300 IF Q$<>"Y" THEN 210
310 PRINT#1,NO,NAM$;",",JYU$;",",TEL$
320 GOSUB370
330 LOCATE 0,18
335 INPUT"ﾂﾂﾞｹﾏｽｶ? Y/N   ",Q$
340 IF Q$="Y" THEN NO=NO+1
350 IF Q$="N" THEN GOSUB 370:CLOSE:GOTO 40
360 GOTO 210
370 LOCATE 0,18
375 PRINT STRING$(40,&H20);:RETURN
380 LOCATE 0,5
381 PRINT STRING$(40,&H20);
382 PRINT STRING$(40,&H20);
383 PRINT STRING$(40,&H20);
385 PRINT STRING$(40,&H20);:RETURN
```

〈プログラムの解説〉

・170～195行　メッセージを画面に出力し（画面出力1）登録するファイル名を入力します．たとえばSDATAと入力します．

・200 行　　　シーケンシャルファイルを開きます．ここで OPEN 命令が使われています．

OPEN 命令は次のような意味を持っています．

```
OPEN "O", [#]1, "0:" +F$
     |     |       |
     |     |       └──ファイルディスクリプタ
     |     └──ファイル番号
     └──O：output 出力を意味します．
```

この行でファイルディスクリプタは "0：" +F$ となっています．195 行でファイル名に SDATA と入力しましたので実際には OPEN "O"，1，"0：SDATA" となり，ドライブ 0 のフロッピィに SDATA とファイル名をつけたファイルをオープンしたことになります．

・220～240 行　　ここで記録するデータを入力しています．

・250～280 行　　入力したデータを画面に表示します(画面出力 2)．

・290～300 行　　入力したデータにまちがいがないかをきいています．

・310 行　　　まちがいがなければファイルにデータを書込んでいます．

```
310  PRINT#1, NO, NAM$; ",", JYU$; ",", TEL$
```

この 310 行では 1 行の PRINT # 文ですべてのデータを記録しています．

文字変数と文字変数の間にデータの区切り記号として "，(コンマ)" を出力していますが，これがないと，データを読込む INPUT # 文を実行したときに最初の文字変数へ連続して読込んでしまいます．

(例)　INPUT#1, NO, NAM$, JYU$, TEL$

区切り符号の "，" がないときは NAM$ にその後の JYU$，TEL$ まで読込んでしまいます．

文字変数を，"，" で区切って PRINT# 文を実行すると 1 行の PRINT# 文で済みますが，この "，" を出力することにより，"，" 以外に余分にファイル上ではスペースもとるためファイルに無駄が生じます．(例 1)(例 2) を参照して下さい．

・335 行　　入力データがなくなったかをきいています．

・350 行　　入力データがなくなったのでファイルをクローズします．

次にメニュー画面へもどりデータ登録ルーチンでの処理は終了です．CLOSE 命令により，ファイルをクローズしています．ファイルはオープンしたら必ずクローズして下さい．

行番号 310 において，データの区切り方による記録内容のちがいを以下に示します．

(例 1)

```
PRINT#1, NO, NAM$; ",", JYU$; ",", TEL$
```

この例により，ファイルにデータを書込んだときのファイルダンプリストです．

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
0000	20	31	20	20	20	20	20	20	20	20	20	20	20	20	41	41	1 AA
0010	41	41	41	41	41	41	2C	20	20	20	20	20	42	42	42	42	AAAAAA, BBBB
0020	42	42	42	42	42	42	42	2C	20	20	43	43	43	2D	43	43	BBBBBBB, CCC-CC
0030	43	2D	43	43	43	43	0D	0A	20	32	20	20	20	20	20	20	C-CCCC.. 2
0040	20	20	20	20	20	20	44	44	44	44	44	44	44	44	44	2C	DDDDDDDDD,
0050	20	20	20	20	45	45	45	45	45	45	45	45	45	45	45	45	EEEEEEEEEEEE
0060	2C	20	46	46	46	2D	46	46	46	2D	46	46	46	46	0D	0A	, FFF-FFF-FFFF..
0070	20	33	20	20	20	20	20	20	20	20	20	20	20	20	47	47	3 GG
0080	47	47	47	2C	20	20	20	20	20	20	20	20	48	48	48	48	GGG, HHHH
0090	48	48	48	48	48	48	48	48	48	48	2C	20	20	20	20	20	HHHHHHHHHH,
00A0	20	20	20	20	20	20	20	20	49	49	49	2D	49	49	49	2D	III-III-
00B0	49	49	49	49	0D	0A	20	34	20	20	20	20	20	20	20	20	IIII.. 4
00C0	20	20	20	20	4A	4A	4A	4A	4A	4A	4A	4A	4A	4A	4A	4A	JJJJJJJJJJJJ
00D0	4A	4A	2C	20	20	20	20	20	20	20	20	20	20	20	20	20	JJ,
00E0	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	KKKKKKKKKKKKKKKK
00F0	4B	2C	20	20	20	20	20	20	20	20	20	20	4C	4C	4C	4C	K, LLLL

（例２）

PRINT#1, NO；PRINT#1, NA$：PRINT#1, JYU$：PRINT#1, TEL$

４つのPRINT＃に分けてファイルにデータを書込んだときのファイルダンプリストです.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
0000	20	31	20	0D	0A	41	41	41	41	41	41	41	41	0D	0A	42	1 ..AAAAAAAA..B
0010	42	42	42	42	42	42	42	42	42	42	0D	0A	43	43	43	2D	BBBBBBBBBB..CCC-
0020	43	43	43	2D	43	43	43	43	0D	0A	20	32	20	0D	0A	44	CCC-CCCC.. 2 ..D
0030	44	44	44	44	44	44	44	44	0D	0A	45	45	45	45	45	45	DDDDDDDD..EEEEEE
0040	45	45	45	45	45	45	0D	0A	46	46	46	2D	46	46	46	2D	EEEEEE..FFF-FFF-
0050	46	46	46	46	0D	0A	20	33	20	0D	0A	47	47	47	47	47	FFFF.. 3 ..GGGGG
0060	0D	0A	48	48	48	48	48	48	48	48	48	48	48	48	48	48	..HHHHHHHHHHHHHH
0070	0D	0A	49	49	49	2D	49	49	49	2D	49	49	49	49	0D	0A	..III-III-IIII..
0080	20	34	20	0D	0A	4A	4A	4A	4A	4A	4A	4A	4A	4A	4A	4A	4 ..JJJJJJJJJJJ
0090	4A	4A	4A	0D	0A	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	JJJ..KKKKKKKKKKK
00A0	4B	4B	4B	4B	4B	4B	0D	0A	4C	4C	4C	4C	4C	4C	4C	4C	KKKKKK..LLLLLLLL
00B0	4C	4C	4C	4C	0D	0A											LLLL..

ファイルは，ステップ数が増えますが，PRINT＃文を分けてプログラムの記述を行なうと，ファイルの専有量が少なくなります.

〈処理上の画面表示〉

```
♣♣♣ デ゛ータ ノ トウロク ♣♣♣




トウロクスル ファイルメイ ノ ニュウリョク     SDATA
```

画面出力 1

```
♣♣♣ デ゛ータ ノ トウロク ♣♣♣

  1
   AAAAAAAA
      BBBBBBBBBBB
         CCC-CCC-CCCC


デ゛ータノアヤマリ ハ ナイデ゛スカ? Y/N   Y
```

画面出力 2

(3) シーケンシャルファイルからデータを入力する(入力処理)

シーケンシャルファイルからデータを入力するためには，以下の命令が必要です.

・入力したいファイルをオープンする OPEN 命令

・シーケンシャルファイルからデータを読む INPUT#命令

・入力するデータがなくなると，ファイルをクローズする CLOSE 命令

シーケンシャルファイルからデータを入力するためにはこの他LINE INPUT#，INPUT$ などがありますが，ここではINPUT#のみを使用しています.

〈処理内容〉

ファイルをオープンし，データを読取り，読取ったデータを画面に出力します．読取るデータがなくなると，ファイルをクローズしてメニュー画面にもどします.

〈プログラム例〉

```
390 ' デ゛ータ ノ ヨミトリ
400 WIDTH 40,20:LOCATE 0,0
405 PRINT"◆◆◆ デ゛ータ ノ ヨミトリ ◆◆◆"
410 LOCATE 0,18
420 INPUT"ファイルメイ ノ ニュウリョク      ",F$
430 OPEN"I",#1,"0:"+F$
440 IF EOF(1) THEN 520
450 INPUT#1,NO,NAM$,JYU$,TEL$
460 LOCATE 2,5:PRINT NO
470 LOCATE 4,6:PRINT NAM$
480 LOCATE 6,7:PRINT JYU$
490 LOCATE 8,8:PRINT TEL$
500 LOCATE 0,18
501 INPUT"ツキ゛ノデ゛ータ ヲ ヒョウシ゛? Y/N   ",H$
502 IF H$="N" THEN GOTO 520
503 GOSUB 380
510 GOTO 440
520 CLOSE:GOSUB 370:LOCATE 0,18
525 INPUT"メニュー ニ モト゛シマスカ? Y/N    ";Q$
530 IF Q$="Y" THEN 40 ELSE 520
```

〈プログラムの解説〉

・405 行　メッセージを出力します.

・420 行　読取るファイル名を入力します.

・430 行　ファイルをオープンします.

データを記録するときとの違いは“O”が“I”になっています.

```
OPEN "I", #1, "0:" +F$
          └─INPUT：入力の意味です.
```

これは INPUT（入力）を意味しています.

・440行　読取るデータがなくなったかをチェックしています.

```
IF EOF(1) THEN 520
       └─ファイル番号
```

EOF（エンドオブファイル，データの終りを意味しています.）

このEOFによりデータがなくなったかをチェックしています.

EOF＝1（真）データがなくなった.

EOF＝0（偽）データがまだある.

・450行　シーケンシャルファイルからデータを入力します.

INPUT＃1，NO，NAM$，JYU$，TEL$

このコマンドによりシーケンシャルファイルからデータを入力します．PRINT＃のところでシーケンシャルファイルにデータの区切りとして "," を出力していましたが，INPUT＃文でこれはデータの区切りとしてしかみないので "," を入力できません.

INPUT＃文ではデータの区切りとしてはコンマ（，），コロン（：），CRコード，LFコードが使えます．最初の文字が引用符（"）の場合は次の引用符まで文字データとみなします.

●460～510行　入力したデータを画面に表示します.

●520～530行　ファイルをクローズします．メニュー画面にもどすかどうかを確記しています.

〈処理上の画面表示〉

```
◆◆◆ デ゛ータ ノ ヨミトリ ◆◆◆

    1
     AAAAAAAA
        BBBBBBBBBBB
           CCC-CCC-CCCC


ツギ゛ノデ゛ータ ヲ ヒョウジ゛? Y/N   Y
```

画面出力

〈参考プログラム〉

INPUT＃文に対するデータの区切り符号の使い方を示す参考プログラムです．コンマ(，)，コロン(．)をデータとして出力し読取るときは引用符(")で囲むとデータとして使用できます.

```
100 OPEN"O",#1,"0:DATA"
110    A$="ABCDEFGH"
120    B$=CHR$(&H0D):'CRコード
130    C$=CHR$(&H0A):'LFコード
140    D$=CHR$(&H2C):' ,コード
150    E$=CHR$(&H3A):' :コード
160    F$=CHR$(&H22):' "コード
170 PRINT#1,A$;B$;
180 PRINT#1,A$;C$;
190 PRINT#1,A$;D$;
200 PRINT#1,A$;E$;
210  PRINT#1
220 PRINT#1,F$;A$;D$;A$;F$
230 PRINT#1,F$;A$;E$;A$;F$
240 CLOSE#1
250 OPEN"I",#1,"0:DATA"
260 IF EOF(1) THEN 300
270 INPUT#1,A$:PRINT A$
280 GOTO 260
300 CLOSE#1
```

```
Ready
RUN
ABCDEFGH
ABCDEFGH
                ABCDEFGH
ABCDEFGH
ABCDEFGH,ABCDEFGH
ABCDEFGH:ABCDEFGH

Ready
█
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
0000	41	42	43	44	45	46	47	48	0D	41	42	43	44	45	46	47	ABCDEFGH.ABCDEFG
0010	48	0A	41	42	43	44	45	46	47	48	2C	41	42	43	44	45	H.ABCDEFGH,ABCDE
0020	46	47	48	3A	0D	0A	22	41	42	43	44	45	46	47	48	2C	FGH:.."ABCDEFGH,
0030	41	42	43	44	45	46	47	48	22	0D	0A	22	41	42	43	44	ABCDEFGH".."ABCD
0040	45	46	47	48	3A	41	42	43	44	45	46	47	48	22	0D	0A	EFGH:ABCDEFGH"..

LINE INPUT# 文を使うと CR コードまですべてのコードを読込むことができます.

(4) シーケンシャルファイルのデータの変更（変更処理）

シーケンシャルファイルでデータを変更するには OPEN 命令で2つのファイルをオープンします. 1つは変更するファイル，もう1つはそのデータを記録するファイルです. 変更するファイルからデータを読取り変更するデータでなければそのまま別のファイルに書き，変更するデータなら，そのデータを変更し，別のファイルに書込みます. データの読込みが終了したらファイルをクローズします. このままでは変更したファイルのファイル名がちがっていますので，このファイル名を，もとのファイル名に直す必要があります. このため KILL 命令でもとのファイル名を消し，次に NAME 命令で変更したデータの入っているファイルの名前にもとのファイル名をつけることにより，もとのファイルのデータが変更されたことになります.

変更するファイルをオープンする．

SDATA 入力

出力 SDATA1

ダミーのファイル名をつけファイルをオープンする．

データを読込み変更し書込む．データがなくなったらファイルをクローズする

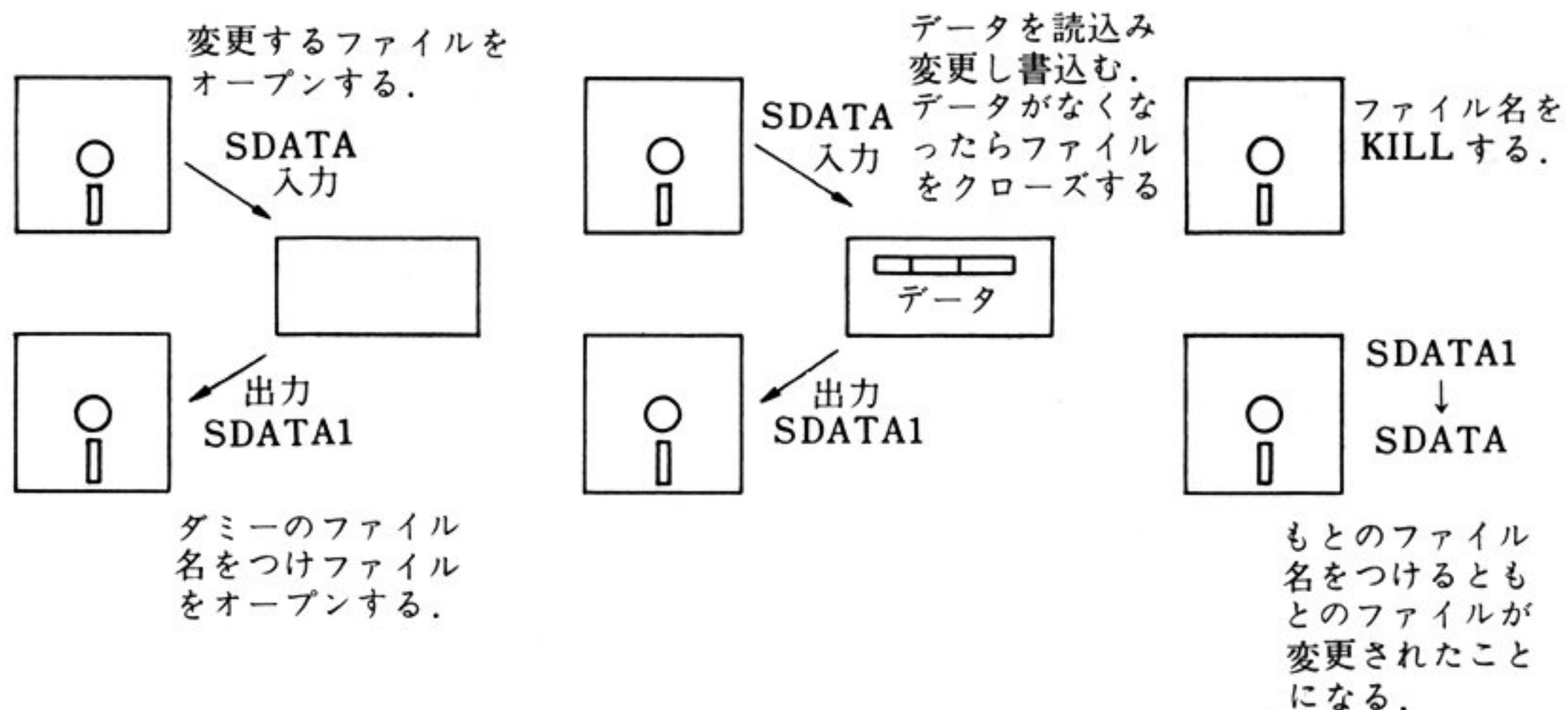
SDATA 入力

データ

出力 SDATA1

ファイル名をKILLする．

SDATA1 ↓ SDATA

もとのファイル名をつけるともとのファイルが変更されたことになる．

データの変更

〈処理内容〉

変更するファイル名と変更するデータの番号を入力します．ファイルをオープンし，データを読取り変更するデータでなければそのまま別のファイルに記録します．

変更データであればそのデータを画面に表示し，変更データを入力します．

変更データの入力がすむとファイルにそのデータを記録します．

入力データがなくなればファイルをクローズしファイル名を変更します．

〈プログラム例〉

```
540 ' ﾃﾞｰﾀ ﾉ ﾍﾝｺｳ
541 CLS:WIDTH 40,20:LOCATE 0,0
542 PRINT"XXX ﾃﾞｰﾀ ﾉ ﾍﾝｺｳ XXX"
550 LOCATE 0,18
555 INPUT"ﾍﾝｺｳｽﾙ ﾌｧｲﾙﾒｲ ﾉ ﾆｭｳﾘｮｸ ";F$
560 LOCATE 0,18
565 INPUT"ﾍﾝｺｳｽﾙ ﾊﾞﾝｺﾞｳ ﾉﾆｭｳﾘｮｸ  ";T
570 OPEN"I",#1,"0:"+F$
580 OPEN"O",#2,"0:"+F$+"1"
590 IF EOF(1) THEN 720
600 INPUT#1,NO,NAM$,JYU$,TEL$
610 IF T=NO THEN 640
620 PRINT#2,NO,NAM$;",",JYU$;",",TEL$
630 GOTO 590
640 LOCATE 2,5:PRINT NO
650 LOCATE 4,6:PRINT NAM$
660 LOCATE 6,7:PRINT JYU$
670 LOCATE 8,8:PRINT TEL$
```

```
680 LOCATE 0,18
685 INPUT"ヘンコウカショ(1:ナマエ 2:ジュウショ 3:TEL)";Q
690 ON Q GOSUB 760,780,800
700 GOSUB 370:LOCATE 0,18
705 INPUT"ヘンコウ シュウリョウ デスカ? Y/N";Q$
710 IF Q$="Y" THEN 620 ELSE 680
720 CLOSE#1:CLOSE#2
730 KILL"0:"+F$
740 NAME"0:"+F$+"1" AS "0:"+F$
750 GOTO 40
760 ' ナマエ ノ ヘンコウ
770 GOSUB 370:LOCATE 0,18
775 INPUT"ナマエ ヲ ニュウリョク    ";NAM$:RETURN
780 ' ジュウショ ノ ヘンコウ
790 GOSUB 370:LOCATE 0,18
795 INPUT"ジュウショ ヲ ニュウリョク ";JYU$:RETURN
800 ' TEL ノ ヘンコウ
810 GOSUB 370:LOCATE 0,18
815 INPUT"TEL ヲ ニュウリョク     ";TEL$:RETURN
```

〈プログラム解説〉

・550～565 行　メッセージを出力しファイル名と変更データの番号を入力する.

・570～580 行　ファイルをオープンします.

　OPEN "I", #1, "0:"+F$　変更するファイル

　OPEN "O", #2, "0:"+F$+"1"　変更したデータを書込むファイル

　ダミーとしてF$に1を加えたファイル名としています.

・590 行　入力データがなくなったかをチェックしています.

・600 行　データを入力します.

・610 行　変更データであるかをチェックしています.

・620 行　データをファイルに書込みます.

・640～670 行　変更するデータを画面に表示します.

・685 行　変更する内容をきいています.

・690 行　変更する内容により分岐します.

・705 行　変更する内容が終りかをきいています.

・720 行　ファイルをクローズします.

・730 行　もとのファイル名を消します.

・740 行　ダミーのファイル名をつけてデータを書込んでおいたファイル名をもとのファイ

ル名にします.

・750行　　メニュー画面にもどります.

・760～775行　　変更する名前を入力します.

・780～795行　　変更する住所を入力します.

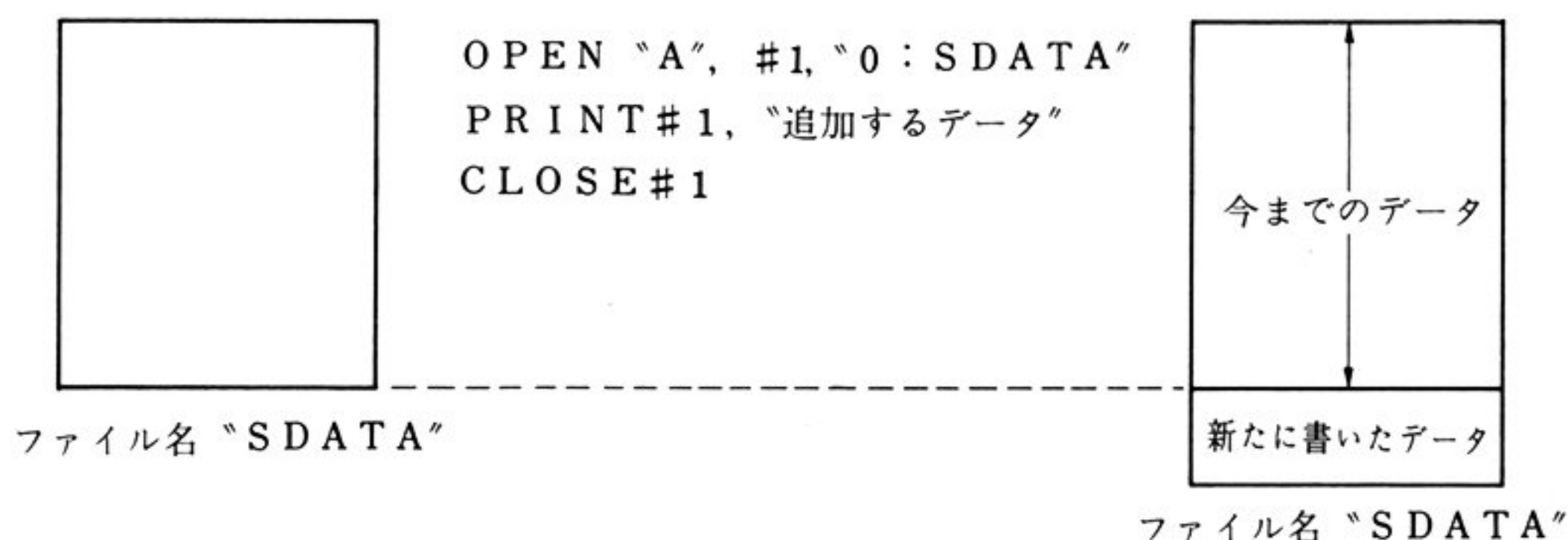
・800～815行　　変更するTELナンバーを入力します.

(5) シーケンシャルファイルのデータの追加(追加処理)

シーケンシャルファイルにデータを追加して書込むには，OPEN命令のときに

OPEN "A", [#]ファイル番号, "ファイルディスクリプタ"

└── 追加指定(シーケンシャルファイルにデータの追加出力を行ないます.)

この追加指定をすれば，今までのファイルのデータに追加して記録します．ただし，指定されたファイル名はフロッピィディスクに存在しなければなりません.

ファイルの追加

〈処理内容〉

追加するファイルを入力し，そのファイルに何個データが入っているかをまず調べます．次に追加するデータを入力し，ファイルに書込みます．追加するデータがなくなれば，ファイルをクローズしてメニュー画面にもどします.

〈プログラム例〉

```
820 ' デ-タ ノ ツイカ
830 CLS:LOCATE 0,0
835 PRINT"♥♣♦♠ デ-タ ノ ツイカ ♥♣♦♠"
840 LOCATE 0,18
845 INPUT"ツイカ スル ファイルメイ ノ ニュウリョク    ";F$
850 OPEN"I",#1,"0:"+F$
860 IF EOF(1) THEN GOTO 890
870 INPUT#1,NO,NAM$,JYU$,TEL$
880 GOTO 860
890 CLOSE#1:GOSUB 370
```

```
900 NO=NO+1
910 OPEN"A",#1,"0:"+F$
920 GOSUB 370
930 LOCATE 0,18
935 INPUT"ﾅﾏｴ ﾉ ﾆｭｳﾘｮｸ          ",NAM$
940 GOSUB 370
950 LOCATE 0,18
955 INPUT"ｼﾞｭｳｼｮ ﾉ ﾆｭｳﾘｮｸ      ",JYU$
960 GOSUB 370
970 LOCATE 0,18
975 INPUT"TEL ﾉ ﾆｭｳﾘｮｸ         ",TEL$
980 GOSUB 370:GOSUB 380
990 LOCATE 2,5:PRINT NO
1000 LOCATE 4,6:PRINT NAM$
1010 LOCATE 6,7:PRINT JYU$
1020 LOCATE 8,8:PRINT TEL$
1030 LOCATE 0,18
1035 INPUT"ﾃﾞｰﾀﾉｱﾔﾏﾘ ﾊ ﾅｲﾃﾞｽｶ? Y/N ",Q$
1040 IF Q$="N" THEN 920
1050 PRINT#1,NO,NAM$;",",JYU$;",",TEL$
1060 GOSUB 370
1070 LOCATE 0,18
1075 INPUT"ﾂﾂﾞｹﾏｽｶ? Y/N  ",Q$
1080 IF Q$="N" THEN GOSUB 370:CLOSE#1:GOTO 40
1090 NO=NO+1:GOTO 920
```

〈プログラム解説〉

- ・830～845 行　メッセージを出力し追加するファイル名を入力します.
- ・850～900 行　ファイルをダミーリードさせ最後のデータのNOだけを取り出します.
  このNOを＋1して次に追加するデータのナンバーにしています.
- ・910 行　ファイルをオープンします.
- ・930～975 行　名前, 住所, TEL NOを入力しています.
- ・990～1020 行　追加するデータを画面に表示します.
- ・1030～1050 行　データにまちがいがなければファイルに書込みます.
- ・1060～1090 行　追加するデータがなければファイルをクローズしてメニュー画面にもどります.
  追加データがまだあれば, 追加データ入力へ分岐します.

〈処理上の画面表示〉

```
♥♣♦♠ テ゛ータ ノ ツイカ ♥♣♦♠

      5
       ABCDEFG
          HIJKLMNOPQRSTU
             VWXYZ

テ゛ータノアトマリ ハ ナイテ゛スカ? Y/N Y
```

出力画面

```
♥♣♦♠ テ゛ータ ノ ツイカ ♥♣♦♠




ツイカ スル ファイルメイ ノ ニュウリョク    ? SDATA
```

出力画面

実際にシーケンシャルファイルに1個データを追加したときのファイルダンプを以下に示します．今までのファイルの後にデータが追加されているのが，おわかりいただけると思います．

〈もとのシーケンシャルファイルのデータ〉

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
0000	20	31	20	20	20	20	20	20	20	20	20	20	20	20	41	41	1 AA
0010	41	41	41	2C	20	20	20	20	20	20	20	20	41	41	41	41	AAA, AAAA
0020	41	41	41	41	41	41	2C	20	20	20	41	41	41	41	41	41	AAAAAA, AAAAAA
0030	41	41	41	41	41	41	41	41	41	0D	0A						AAAAAAAAA..

〈1個データを追加したシーケンシャルファイル〉

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
0000	20	31	20	20	20	20	20	20	20	20	20	20	20	20	41	41	1 AA
0010	41	41	41	2C	20	20	20	20	20	20	20	20	41	41	41	41	AAA, AAAA
0020	41	41	41	41	41	41	2C	20	20	20	41	41	41	41	41	41	AAAAAA, AAAAAA
0030	41	41	41	41	41	41	41	41	41	0D	0A	20	32	20	20	20	AAAAAAAAA.. 2
0040	20	20	20	20	20	20	20	20	20	42	42	42	42	42	2C	20	BBBBB,
0050	20	20	20	20	20	20	20	42	42	42	42	42	42	42	42	42	BBBBBBBBB
0060	42	2C	20	20	20	42	42	42	42	42	42	42	42	42	42	42	B, BBBBBBBBBBB
0070	42	42	42	42	0D	0A											BBBB..

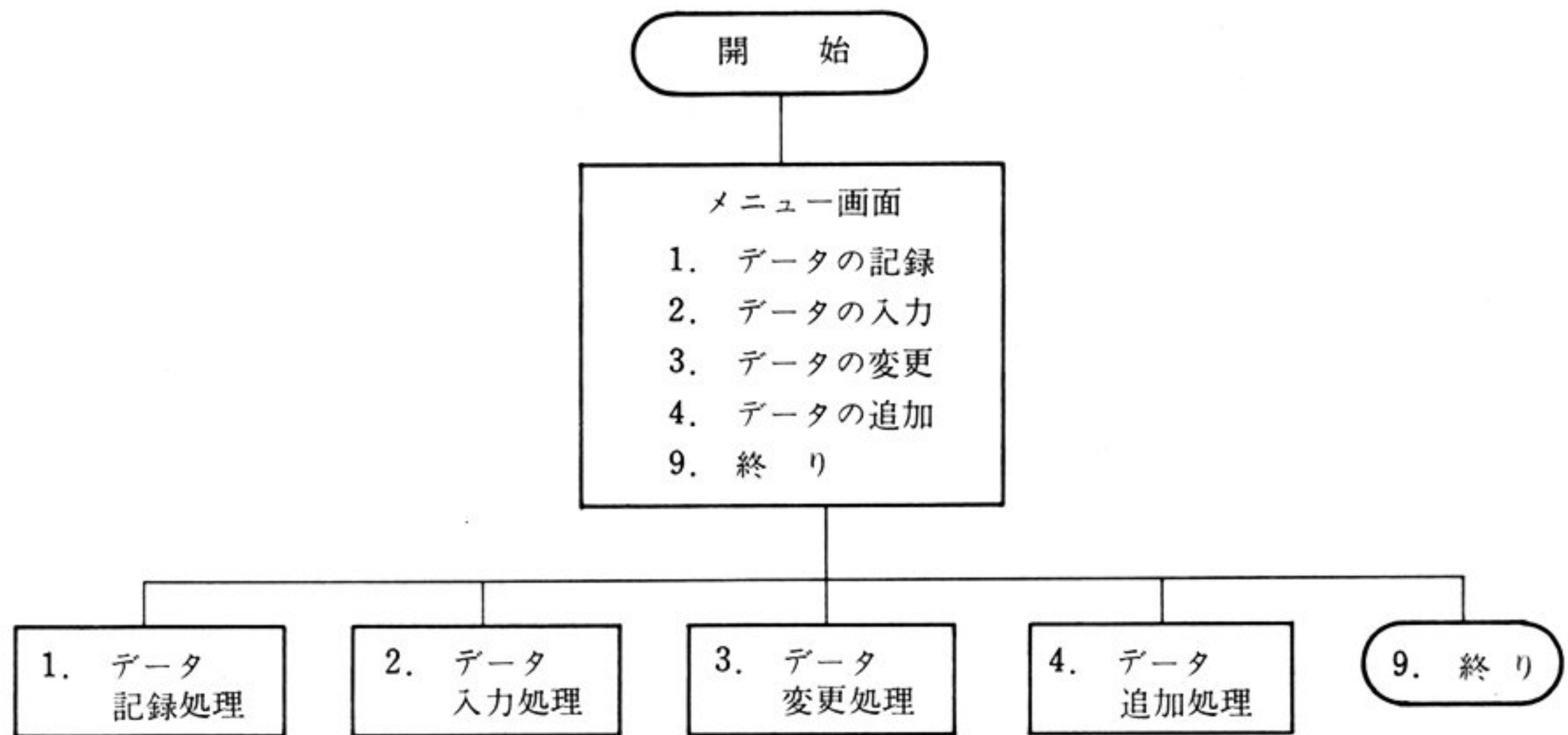
13.6.2 ランダムファイルの扱い方

シーケンシャルファイルにおいて説明を行なったものと同じプログラム例をランダムファイルに対するプログラムとした例を説明しながらランダムファイルに必要なF-BASICの命令等の説明を行ないます．

ランダムファイルを利用したプログラムの処理内容を次に示します．

開　始

メニュー画面
1. データの記録
2. データの入力
3. データの変更
4. データの追加
9. 終　り

1. データ記録処理　2. データ入力処理　3. データ変更処理　4. データ追加処理　9. 終り

プログラムは4つの処理に別れています.

1. データ記録処理　ランダムファイルへデータを記録する.

2. データ入力処理　ランダムファイルよりデータを入力する.

3. データ変更処理　ランダムファイルのデータを変更する.

4. データ追加処理　ランダムファイルへデータを追加する.

(1) メニュー

〈処理内容〉

この処理においては，ディスプレイへ，メッセージを出力して，目的の処理の選択を行ない，その処理へ実行を移します.

〈プログラム例〉

```
10 'ｼﾞｭｳｼｮﾛｸ ﾌﾟﾛｸﾞﾗﾑ
20 'ﾒﾆｭｰｶﾞﾒﾝ
30 '
40 WIDTH 40,20
50 SYMBOL(0,0),"♥♥♥ ｼﾞｭｳｼｮﾛｸ♥♥♥",4,2,4
55 SYMBOL(60,20),"(ﾗﾝﾀﾞﾑﾌｧｲﾙ)",4,2,2
60 LOCATE 4,4:PRINT" 1: ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ"
70 PRINT"      2: ﾃﾞｰﾀ ﾉ ﾖﾐﾄﾘ"
80 PRINT"      3: ﾃﾞｰﾀ ﾉ ﾍﾝｺｳ"
90 PRINT"      4: ﾃﾞｰﾀ ﾉ ﾂｲｶ"
100 PRINT"      9: ｵﾜﾘ"
110 INPUT"ｼｮﾘﾊﾞﾝｺﾞｳ ﾉ ﾆｭｳﾘｮｸ      ",Q
120 ON Q GOTO 160,390,540,830
130 INPUT"ｼｮﾘ ｦ ｵﾜﾘ ﾏｽｶ? Y/N      ",Q$
140 IF Q$<>"Y" THEN GOTO 150
```

```
144 GOSUB 370
145 LOCATE 0,12:PRINT"♠♠♠ オワリ ♠♠♠"
146 END
150 GOTO 40
```

〈プログラムの解説〉

・ 40～100行　　メッセージを出力しています.

・110行　　目的の処理を選択します.

・120行　　目的の処理へ実行を移します.

・130～150行　　プログラムを終了させるかの判定を行ないます.

〈処理上の画面表示〉

```
♥♥♥ ジュウショロク♥♥♥
(ランダムファイル)
    1: データ ノ トウロク
    2: データ ノ ヨミトリ
    3: データ ノ ヘンコウ
    4: データ ノ ツイカ
    9: オワリ
ショリバンゴウ ノ ニュウリョク    1
```

出力画面

(2) ランダムファイルにデータを記録する (データ記録処理)

ランダムファイルにデータを記録するためには, 以下の命令が必要です.

・書込むファイルに名前をつけファイルを開く OPEN "R" 命令

・データをファイルに記録する PUT# 命令

・ファイルを閉じる CLOSE 命令

この3つの命令により構成されます.

〈処 理 内 容〉

登録するファイル名を入力し, 次にFIELD文でランダムバッファの変数の領域を設定します.

次にデータを入力し, まちがいがなければランダムファイルにデータの記録を行ないます. 入力するデータがなくなると, ファイルを閉じてメニュー画面にもどります.

〈プログラム例〉

```
160 ' ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ(R)
170 WIDTH 40,20
180 NO=1:PRINT"♣♣♣ ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ(R)  ♣♣♣"
190 LOCATE 0,18
195 INPUT"ﾄｳﾛｸｽﾙ ﾌｧｲﾙﾒｲ ﾉ ﾆｭｳﾘｮｸ   ",F$
200 OPEN"R",#1,"0:"+F$:I=1
201 FIELD#1,4ASNO$,20ASNAM$,30ASJYU$,18ASTEL$
205 N$=MKS$(I):LSET NO$=N$
210 GOSUB 370:LOCATE 0,18
220 INPUT"ｼﾒｲ ﾉ ﾆｭｳﾘｮｸ   ",NA$
221 LSET NAM$=NA$
225 GOSUB 370:LOCATE 0,18
230 INPUT"ｼﾞｭｳｼｮ ﾉ ﾆｭｳﾘｮｸ   ",JY$
231 LSET JYU$=JY$
235 GOSUB 370:LOCATE 0,18
240 INPUT"ﾃﾞﾝﾜﾊﾞﾝｺﾞｳ ﾉ ﾆｭｳﾘｮｸ   ",TE$
241 LSET TEL$=TE$
245 GOSUB 380:LOCATE 2,5
250 PRINT I
260 LOCATE 4,6:PRINT NA$
270 LOCATE 6,7:PRINT JY$
280 LOCATE 8,8:PRINT TE$
290 LOCATE 0,18
295 INPUT"ﾃﾞｰﾀﾉｱﾔﾏﾘ ﾊ ﾅｲﾃﾞｽｶ? Y/N  ",Q$
300 IF Q$<>"Y" THEN 205
310 PUT#1,I:' 1ﾚｺｰﾄﾞ ﾉ ｷﾛｸ
320 GOSUB370
330 LOCATE 0,18
335 INPUT"ﾂﾂﾞｹﾏｽｶ? Y/N   ",Q$
340 IF Q$="Y" THEN I=I+1:GOTO 205
350 IF Q$="N" THEN GOSUB 370:CLOSE:GOTO 40
360 GOTO 320
370 LOCATE 0,18
375 PRINT STRING$(40,&H20);:RETURN
380 LOCATE 0,5
381 PRINT STRING$(40,&H20);
382 PRINT STRING$(40,&H20);
383 PRINT STRING$(40,&H20);
385 PRINT STRING$(40,&H20);:RETURN
```

〈プログラム解説〉

・170～195 行　メッセージを出力し登録するファイル名を入力します.

・200 行　ランダムファイルをオープンします.

```
OPEN "R", [#]1, "0:" +F$
```

- "0:" +F$ ―― ファイルディスクリプタ
- [#]1 ―― ファイル番号
- "R" ―― ランダムファイルの入出力を行ないます. 指定されたファイル名がディスク上に存在しないときは, そのファイルが新たに登録されます.

たとえば, ファイル名を RDATA と入力しますと実際は OPEN "R", #1, "0:RDATA" となります.

・201 行　FIELD 文でランダムバッファの変数の領域を割当てます.

ここでは NO$（ナンバー）を 4 桁, NAM$（名前）を 20 桁, JYU$（住所）を 30 桁, TEL$（TEL NO）を 18 桁としています. ランダムファイルはセクタ単位でデータを処理します.

このプログラムでは 72 バイトしか使っていません.

1 セクタは 256 バイトですから残りの 184 バイトは空です. このように短いデータをランダムファイルで処理すると, ファイルに無駄が生じます. 実際のプログラムでランダムファイルを使うときはファイルに無駄が生じないようにすることが大事です.

・205 行　NO$（ナンバー）にデータを入力します.

N$=MKS$(I):LSET NO$=N$

LSET でランダムファイルに数値を入れるときはすべて文字に変換する必要があるので MKS$ で単精度形式の数値を 4 バイトの文字列に変換し, LSET でデータをランダムバッファに移します.

ここではナンバー（I）を一度 N$ に MKS$ で文字変換し LSET でランダムバッファの NO$ に移しています.

・220～241 行　名前, 住所, TEL NO を入力して, そのデータをすべて LSET によりランダムバッファに移します.

・250～280 行　入力したデータを表示します.

・290～310 行　入力データにあやまりがなければランダムファイルにデータを記録します.

```
PUT#1, I
```

- I ―― レコード番号
- PUT#1 ―― ファイル番号

ランダムファイルに対する出力命令です.

この PUT 文によりランダムファイルにバッファの内容を記録します．レコード番号を省略すると，直前にそのファイルに対して PUT または GET した次のレコードに出力されます．

レコード番号は1～最大レコード番号（残りクラスタ数×8）まで指定できます．

・320～350 行　入力データがなくなったらファイルをクローズしてメニュー画面へもどし，まだ入力データがあるときは，データの入力の行へ分岐します．

```
♣♣♣ デ゛ータ ノ トウロク(R) ♣♣♣



トウロクスル ファイルメイ ノ ニュウリョク     RRR
```

出力画面 1

```
♣♣♣ デ゛ータ ノ トウロク(R) ♣♣♣

    1
     AAAAAA
        BBBBBBBBB
           CCC-CCC-CCCC


デ゛ータノアヤマリ ハ ナイデ゛スカ? Y/N   Y
```

出力画面 2

(3) ランダムファイルからデータを入力する（入力処理）

ランダムファイルからデータを入力するためには，以下の命令が必要です．

- 読取りたいファイルをオープンする OPEN〝R″命令
- ランダムファイルからデータを読む GET 命令
- 読取るデータがなくなったらファイルをクローズする CLOSE 命令

これらの命令で構成されます．

〈処 理 内 容〉

ファイルをオープンし，FIELD 文でランダムバッファの変数の領域を設定します．

次にデータを入力し画面に表示します．読取るデータがなくなると，ファイルをクローズし，再度実行しないときはメニュー画面にもどります．

〈プログラム例〉

```
390 ' デ゛ータ ノ ヨミトリ(R)
400 WIDTH 40,20:LOCATE 0,0
405 PRINT"◆◆◆ デ゛ータ ノ ヨミトリ(R) ◆◆◆"
410 LOCATE 0,18
420 INPUT"ファイルメイ ノ ニュウリョク     ",F$
430 OPEN"R",#1,"0:"+F$
431 FIELD#1,4ASNO$,20ASNAM$,30ASJYU$,18ASTEL$
432 LAG=LOF(1)
```

```
440 FOR I=1 TO LAG
450 GET#1,I:' 1ﾚｺｰﾄﾞ ﾉ ﾖﾐﾄﾘ
451 N=CVS(NO$)
460 LOCATE 2,5:PRINT N
470 LOCATE 4,6:PRINT NAM$
480 LOCATE 6,7:PRINT JYU$
490 LOCATE 8,8:PRINT TEL$
500 LOCATE 0,18
501 INPUT"ﾂｷﾞﾉﾃﾞｰﾀ ｦ ﾋｮｳｼﾞ? Y/N  ",H$
502 IF H$<>"Y" THEN 520
503 GOSUB 380
510 NEXT
520 CLOSE:GOSUB 370:LOCATE 0,18
525 INPUT"ﾒﾆｭｰ ﾆ ﾓﾄﾞｼﾏｽｶ? Y/N   ";Q$
530 IF Q$="Y" THEN 40 ELSE 520
```

〈プログラム解説〉

・400～420 行　メッセージを画面に出力し読取るファイル名を入力します.

・430 行　ファイルをオープンします.
ランダムファイルの入出力はOPEN "R" により行なえ，シーケンシャルファイルのときのOPEN命令のように入力と出力の区別はありません.

・431 行　FIELD文でランダムバッファの変数の領域を割当てます.
ここではNO$（ナンバー）を4桁，NAM$（名前)20桁，JYU$（住所）を30桁，TEL$（TEL NO）を18桁としています.

・432 行　オープンしたランダムファイルが何レコードまで使用しているかチェックしています.

LAG=LOF(1)

ランダムファイルに対してLOF関数を用いると，ランダムファイルの最大レコード番号が返されます.

・440 行　1レコードから最大レコード番号まで読取りをくり返すためのFOR文です.

・450 行　ランダムファイルからデータを読取り，ランダムバッファへ格納します.
GET [#] ファイル番号 [, レコード番号]
ランダムファイルから指定されたレコードをランダムバッファに読取ります.
レコード番号は何番目のレコードを読取るかを示し，その値は1～最大レコード番号までです．レコード番号の指定がないときは，直前にそのファイルに対してGETまたはPUTした次のレコードが読込まれます.

・451 行　ランダムファイルから読出した文字を数値に変換します.

N＝CVS (NO$)　4バイトの文字列を単精度形式数値に変換します.

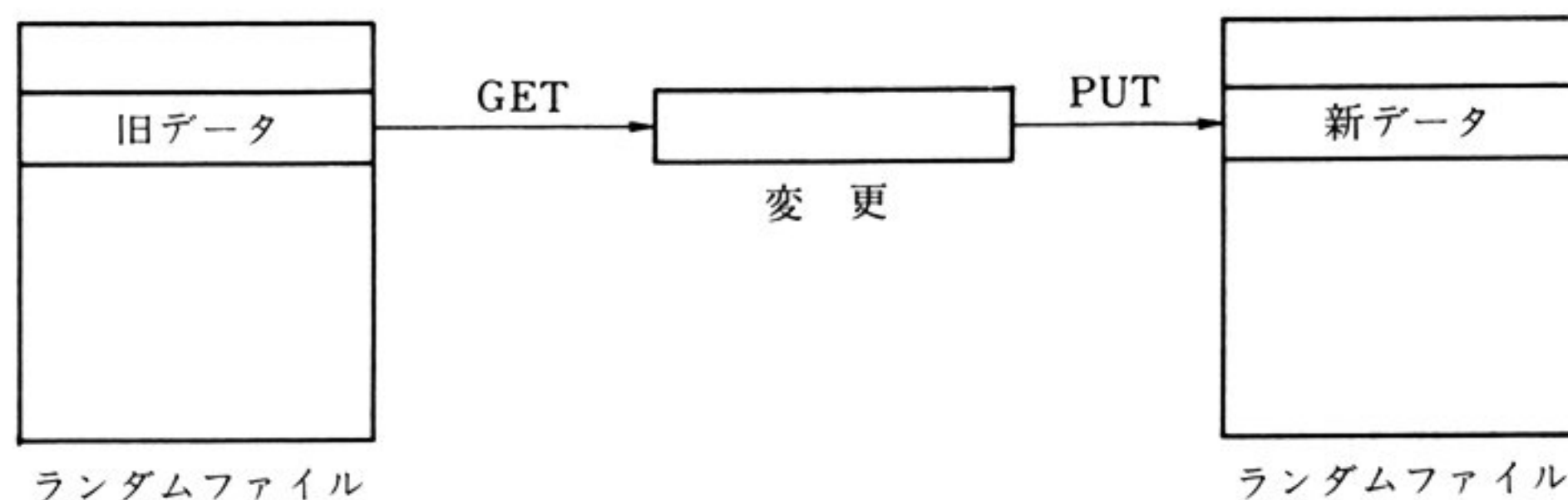
このほか整数のときはCVI（2バイトの文字列を整数に変換），倍精度のときはCVD（8バイトの文字列を倍精度形式の数値に変換）を用います.

・460～503 行　読出したデータを画面に表示します.

・520 行　データの読取りを終了し，ファイルをクローズします.

・525～530 行　メニュー画面にもどします.

（4） ランダムファイルのデータを変更する（変更処理）

ランダムファイルは，レコード番号でデータの入出力ができますので，変更するデータが記録されたレコードをGET文で読取り変更箇所を入力し，まちがいがなければPUT文で同一のレコードへ記録すればデータの変更ができます.

〈処 理 内 容〉

ファイルをオープンし，FIELD文でランダムバッファの変数の領域を設定します.

変更するレコード番号を入力し，GET文で読取ります．読取ったデータを画面に表示し，変更するデータを入力してまちがいがなければPUT文でデータを記録します．次にファイルをクローズしてメニュー画面にもどします.

〈プログラム例〉

```
540 ' デ゛ータ ノ ヘンコウ(R)
541 CLS:WIDTH 40,20:LOCATE 0,0
542 PRINT"XXX デ゛ータ ノ ヘンコウ(R)  XXX"
550 LOCATE 0,18
555 INPUT"ヘンコウスル ファイルメイ ノ ニュウリョク ";F$
560 OPEN"R",#1,"0:"+F$
565 FIELD#1,4ASNO$,20ASNAM$,30ASJYU$,18ASTEL$
570 LAG=LOF(1):GOSUB 370
580 LOCATE 0,18
590 INPUT"ヘンコウスル バ゛ンゴ゛ウ ノ ニュウリョク  ";I
600 IF I<1 OR I>LAG THEN 590
610 GET#1,I:' 1レコート゛ ノ ヨミトリ
620 N=CVS(NO$)
630 GOSUB 380
```

```
640 LOCATE 2,5:PRINT N
650 LOCATE 4,6:PRINT NAM$
660 LOCATE 6,7:PRINT JYU$
670 LOCATE 8,8:PRINT TEL$
680 LOCATE 0,18
685 INPUT"ﾍﾝｺｳｶｼｮ(1:ﾅﾏｴ 2:ｼﾞｭｳｼｮ 3:TEL)";Q
690 ON Q GOSUB 760,780,800
695 PUT#1,I:' 1ﾚｺｰﾄﾞ ﾉ ｷﾛｸ
700 GOSUB 370:LOCATE 0,18
705 INPUT"ﾍﾝｺｳ ｼｭｳﾘｮｳ ﾃﾞｽｶ? Y/N";Q$
710 IF Q$<>"Y" THEN GOSUB 370:GOTO 580
720 CLOSE#1
750 GOTO 40
760 ' ﾅﾏｴ ﾉ ﾍﾝｺｳ
770 GOSUB 370:LOCATE 0,18
775 INPUT"ﾅﾏｴ ｦ ﾆｭｳﾘｮｸ      ";NA$
776 LSET NAM$=NA$:RETURN
780 ' ｼﾞｭｳｼｮ ﾉ ﾍﾝｺｳ
790 GOSUB 370:LOCATE 0,18
795 INPUT"ｼﾞｭｳｼｮ ｦ ﾆｭｳﾘｮｸ ";JY$
796 LSET JYU$=JY$:RETURN
800 ' TEL ﾉ ﾍﾝｺｳ
810 GOSUB 370:LOCATE 0,18
815 INPUT"TEL ｦ ﾆｭｳﾘｮｸ      ";TE$
816 LSET TEL$=TE$:RETURN
```

〈プログラム例〉

- ・541～555 行　メッセージ画面を出力してファイル名を入力します.
- ・560 行　ファイルをオープンします.
- ・565 行　FIELD 文でランダムバッファの変数の領域を設定します.
  ここでは NO$（ナンバー）を 4 桁, NAM$（名前）20 桁, JYU$（住所）を 30 桁, TEL$（TEL NO）を 18 桁に設定しています.
- ・570 行　ランダムファイルが何レコードまでデータが入っているかを調べます.
- ・580～600 行　何番目のレコードを変更するかを入力し, そのレコードが 1 より小さいか, 使用レコード数より大きいかを判定しています.
- ・610～670 行　ランダムファイルからデータを読取り画面に表示します.
- ・680～690 行　どの箇所を変更するかを入力し, その内容により各変更ルーチンへ処理を移します.
- ・695 行　変更データをランダムファイルへ記録します.
- ・700～750 行　その他に変更する箇所がなければメニュー画面にもどります.
- ・775～776 行　変更する名前を入力し LSET 文でランダムバッファに移します.
- ・795～796 行　変更する住所を入力し LSET 文でランダムバッファに移します.

・815～816行　　変更するTEL NOを入力しLSET文でランダムバッファに移します.

(5) ランダムファイルにデータを追加する（追加処理）

ランダムファイルにデータを追加するにはLOF関数で，データを追加するランダムファイルが何レコードまで使用しているかを調べます．そのレコード数に＋1したレコード番号から，PUT文でレコード番号を指定してデータを記録すればランダムファイルのデータの追加ができます.

データ追加前のランダムファイル
1
2
3
4
5
6

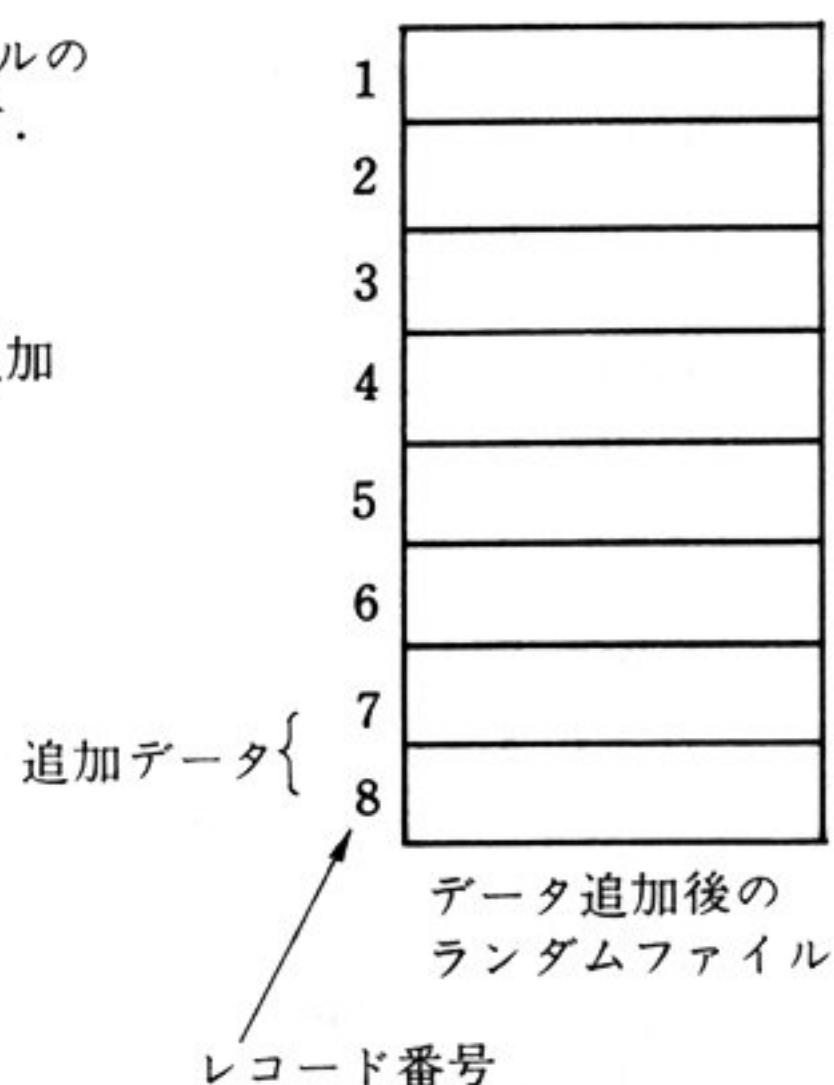
レコード番号

LOF関数でランダムファイルの最大レコード番号を調べます.
この図では6ですから

6＋1＝7

7レコード以降にデータが追加されます.

データ追加後のランダムファイル
1
2
3
4
5
6
7 (追加データ)
8 (追加データ)

レコード番号

ランダムファイルのデータの追加

〈処理内容〉

ファイルをオープンし，FIELD文でランダムバッファの変数の領域を設定します.

LOF関数でランダムファイルの最大レコード数を調べ追加データを入力し，まちがいがなければPUT文でランダムファイルにデータを追加していきます．追加するデータがなくなればファイルをクローズしてメニュー画面にもどします.

〈プログラム例〉

```
820 ' デ゛ータ ノ ツイカ(R)
830 CLS:LOCATE 0,0
835 PRINT"♥♣♦♠ デ゛ータ ノ ツイカ(R) ♥♣♦♠"
840 LOCATE 0,18
845 INPUT"ツイカ スル ファイルメイ ノ ニュウリョク    ";F$
850 OPEN"R",#1,"0:"+F$
860 FIELD#1,4ASNO$,20ASNAM$,30ASJYU$,18ASTEL$
870 I=LOF(1):I=I+1
880 GOSUB 370:LOCATE 0,18
890 PRINT"ツイカノ デ゛ータ ハ ";I;"バ゛ン イコウテ゛ス。"
900 FOR J=1TO 2000:NEXT
920 GOSUB 370
930 LOCATE 0,18
935 INPUT"ナマエ ノ ニュウリョク            ",NA$
```

```
936 LSET NAM$=NA$
940 GOSUB 370
950 LOCATE 0,18
955 INPUT"ｼﾞｭｳｼｮ ﾉ ﾆｭｳﾘｮｸ      ",JY$
956 LSET JYU$=JY$
960 GOSUB 370
970 LOCATE 0,18
975 INPUT"TEL ﾉ ﾆｭｳﾘｮｸ          ",TE$
976 LSET TEL$=TE$
980 GOSUB 370:GOSUB 380
985 N$=MKS$(I):LSET NO$=N$
990 LOCATE 2,5:PRINT I
1000 LOCATE 4,6:PRINT NAM$
1010 LOCATE 6,7:PRINT JYU$
1020 LOCATE 8,8:PRINT TEL$
1030 LOCATE 0,18
1035 INPUT"ﾃﾞｰﾀﾉｱﾔﾏﾘ ﾊ ﾅｲﾃﾞｽｶ? Y/N ",Q$
1040 IF Q$="N" THEN 920
1050 PUT#1,I:' 1ﾚｺｰﾄﾞ ﾉ ｷﾛｸ
1055 I=I+1
1060 GOSUB 370
1070 LOCATE 0,18
1075 INPUT"ﾂﾂﾞｹﾏｽｶ? Y/N  ",Q$
1080 IF Q$="N" THEN GOSUB 370:CLOSE#1:GOTO 40
1090 GOTO 920
```

〈プログラム解説〉

・830～845行　メッセージを画面に出力し追加するファイル名を入力します.

・850行　ファイルをオープンします.

・860行　FIELD文でランダムバッファの変数の領域を設定します.
ここではNO$（ナンバー）4桁，NAM$（名前）20桁，JYU$（住所）を30桁，TEL$（TEL NO）を18桁に設定しています.

・870～900行　ランダムファイルが何レコードまで使用しているかを調べます.
この値に＋1したレコードから追加データが記録されます.

・930～985行　追加データを入力し，LSET文でランダムバッファにデータを移します.

・990～1020行　追加データを画面に表示します.

・1030～1055行　追加データにまちがいがなければPUT文でデータを記録しレコード番号を＋1します.

・1060～1090行　追加するデータがなくなればファイルをクローズしてメニュー画面にもどります.

13.7 MERGE 命令と CHAIN 命令の使い方

13.7.1 MERGE(マージ)

メモリ上のプログラムにフロッピィディスク上のプログラムファイルの対応する行番号を置きかえます．マージされるプログラムファイルは，アスキー形式でなければなりません．

MERGE "　　　　" [, R]

- "　　" ：ファイルディスクリプタの指定をします．
- [, R] ：マージした後，プログラムの先頭から実行を開始する指定で，省略はマージだけです．

以下にマージを行なうプログラム例を示して，マージについて説明します．

メモリ上のプログラム

```
5 ''ｺﾉｷﾞｮｳ ﾊ ｺﾒﾝﾄﾃﾞｽ
10 A$="FUJITSU"
20 B$="MICRO 7"
30 PRINT A$;" ";B$
40 MERGE "ｺﾄﾞﾓ",R
```

ミニフロッピィディスク上のマージされるプログラム（ファイル名：コドモ）

```
5 GOTO 40
40 PRINT A$;" ";B$
50 PRINT "MERGE END"
```

メモリ上のプログラムの実行を行なうと，メモリ上のプログラムの内容は，以下に示すリストのようになります．

```
5 GOTO 40
10 A$="FUJITSU"
20 B$="MICRO 7"
30 PRINT A$;" ";B$
40 PRINT A$;" ";B$
50 PRINT "MERGE END"
```

- 5, 40 ← "コドモ" プログラムの内容と置き換えられます．
- 50 ← "コドモ" プログラムの内容が追加されます．

実行を行なったディスプレイへの表示を以下に示します．

```
Ready
RUN
FUJITSU MICRO 7

MERGE END

Ready
```

- FUJITSU MICRO 7 ← マージ前のメモリ上のプログラムの行番号 30 による出力です．
- 空行 ← マージ後のメモリ上のプログラムの行番号40による出力ですが，変数が渡されませんので，何も出力されず，改行のみ行ないます．
- MERGE END ← マージ後のメモリ上のプログラムでの行番号50による出力です．

13.7.2 CHAIN(チェイン)・COMMON(コモン)

CHAIN 文は，プログラムの連結とそれにともない，変数を渡します．また，CHAIN 文には，プログラムのマージ（MERGE）を行なう機能と，プログラムを任意の行番号からスタートさせる機能および，プログラムの任意の行番号を消去する機能があります．

CHAIN	MERGE	"ファイルディスクリプタ"	,	行番号	,	ALL	,	DELETE範囲
	①	②		③		④		⑤

① MERGE（マージ）

メモリ上のプログラムと，ファイルディスクリプタにより指定を行なったプログラムファイルとをマージする場合に指定します．

前述の MERGE 命令とマージする方法は同様ですが，マージするプログラムは，アスキー形式だけではなく，バイナリ形式であってもかまいません．ただし，バイナリ形式のプログラムファイルをマージする場合，そのプログラムファイルの先頭の行番号は，メモリ上のプログラムの最大行番号より大きくなければなりません．

② ファイルディスクリプタ

ミニフロッピィディスク上のプログラムファイルを指定します．ドライブ0を指定する場合は，デバイス名を省略できます．

③ 実行開始行番号

チェイン後のプログラムを実行する行番号を指定します．

省略した場合は，最初の行番号より実行を開始します．

④ ALL（オール）

変数をすべてチェインした後のプログラムへ渡す場合，"ALL" を記述します．省略した場合，渡す変数は，COMMON 文により，指定しなければなりません．COMMON 文も省略すると，変数を渡すことはできません．

⑤ DELETE（デリート）範囲

チェインを行なう際，メモリ上のプログラムにおける，不要な行番号を消去するための指定を行ないます．

チェインは，指定した行番号の消去を行なった後，実行します．

COMMON　変数名

COMMON文は，CHAIN命令とともに利用され，CHAIN命令が実行された時，指定を行なった変数をチェインするプログラムへ渡します．

COMMONの記述例

```
COMMON  A, B, C$, D$, E( ) .F$( )
```

変数名は，〝，〃（コンマ：comma）により区切り，指定します．〝E(　)〃は，DIM命令により，配列を宣言した変数名です．

では次に，CHAIN，COMMON文を利用したプログラム例を以下に示します．

CHAIN文を使用したサンプルプログラム例

```
10 ' ﾌﾟﾛｸﾞﾗﾑ A
20 INPUT"ﾜﾀｽ ﾍﾝｽｳ ﾉ ﾆｭｳﾘｮｸ",HENSU$
30 INPUT"ﾁｪｲﾝｽﾙ ﾌﾟﾛｸﾞﾗﾑ B OR C",Q$
40 IF Q$="B"THEN 70
50 IF Q$="C"THEN 90
60 GOTO 30
70 CHAIN MERGE"0:B",200,ALL,DELETE200-
80 '
90 CHAIN MERGE"0:C",200,ALL,DELETE200-
```

```
200 ' ﾌﾟﾛｸﾞﾗﾑ B
210 PRINT"ﾌﾟﾛｸﾞﾗﾑ B"
220 PRINT HENSU$
230 GOTO 20
```

```
200 ' ﾌﾟﾛｸﾞﾗﾑ C
210 PRINT"ﾌﾟﾛｸﾞﾗﾑ C"
220 PRINT HENSU$
230 GOTO 20
```

<プログラム説明>

プログラムは，A，B，Cの三種類あります．まずプログラムAを実行させ，引き渡す変数を入力した後，その変数を引き渡すプログラムを（1）か（2）で選択します．

（1） を入力するとプログラムBが，プログラムAとマージされます．

（2） を入力するとプログラムCが，プログラムAとマージされます．また，マージされた，プログラムは，引き渡された変数を表示し再びプログラムAから実行します．

<プログラムAとプログラムBがマージされたメモリ上のプログラム>

```
10 ' ﾌﾟﾛｸﾞﾗﾑ A
20 INPUT"ﾜﾀｽ ﾍﾝｽｳ ﾉ ﾆｭｳﾘｮｸ",HENSU$
30 INPUT"ﾁｪｲﾝｽﾙ ﾌﾟﾛｸﾞﾗﾑ B OR C",Q$
40 IF Q$="B"THEN 70
50 IF Q$="C"THEN 90
60 GOTO 30
70 CHAIN MERGE"0:B",200,ALL,DELETE200-
80 '
90 CHAIN MERGE"0:C",200,ALL,DELETE200-
200 ' ﾌﾟﾛｸﾞﾗﾑ B
210 PRINT"ﾌﾟﾛｸﾞﾗﾑ B"
```

```
220 PRINT HENSU$
230 GOTO 20
```

〈プログラムAとプログラムCがマージされたメモリ上のプログラム〉

```
10 ' ﾌﾟﾛｸﾞﾗﾑ A
20 INPUT"ﾜﾀｽ ﾍﾝｽｳ ﾉ ﾆｭｳﾘｮｸ",HENSU$
30 INPUT"ﾁｪｲﾝｽﾙ ﾌﾟﾛｸﾞﾗﾑ B OR C",Q$
40 IF Q$="B"THEN 70
50 IF Q$="C"THEN 90
60 GOTO 30
70 CHAIN MERGE"0:B",200,ALL,DELETE200-
80 '
90 CHAIN MERGE"0:C",200,ALL,DELETE200-
200 ' ﾌﾟﾛｸﾞﾗﾑ C
210 PRINT"ﾌﾟﾛｸﾞﾗﾑ C"
220 PRINT HENSU$
230 GOTO 20
```

次にCHAINとCOMMONを使用した例を下記に示します．また，ここではマージ機能と〝ALL″の機能は，使用していません．

CHAINとCOMMONのサンプルプログラムの実行出力

```
Ready
RUN"0:PROG-A"
ﾌﾟﾛｸﾞﾗﾑ A
ｱﾅﾀ ﾉ ﾅﾏｴ ﾊ?    TATEKAWA
ｱﾅﾀ ﾉ ﾈﾝﾚｲ ﾊ?   30
ﾌﾟﾛｸﾞﾗﾑ B
ｱﾅﾀ ﾊ TATEKAWA ｻﾝ ﾃﾞｽﾈ
ｿｼﾃ ｱﾅﾀ ﾊ  30  ｻｲ ﾃﾞｽﾈ
ﾌﾟﾛｸﾞﾗﾑ A
ｱﾅﾀ ﾉ ﾅﾏｴ ﾊ?
```

〈プログラム説明〉

プログラムはAとBの二種類あります．まずプログラムAを実行させ，プログラムBに引き渡す変数を，COMMONにより指定します．そして，変数をそれぞれ入力した後，DELETE機能によりプログラムAを消します．さらに，プログラムBをメモリ上にチェインし，プログラムBは，引き渡された変数を表示した後，再びプログラムAをメモリ上にチェインします．

CHAINとCOMMONのサンプルプログラムリスト

```
10 ' CHAIN ﾄ COMMON ﾉ ﾌﾟﾛｸﾞﾗﾑﾚｲ
20 ' ﾌﾟﾛｸﾞﾗﾑ A(PROB-A)
30 PRINT"ﾌﾟﾛｸﾞﾗﾑ A"
40 INPUT"ｱﾅﾀ ﾉ ﾅﾏｴ ﾊ?     ",NAM$
50 INPUT"ｱﾅﾀ ﾉ ﾈﾝﾚｲ ﾊ?    ",OLD
60 COMMON NAM$,OLD
70 CHAIN "0:PROG-B",10
```

```
10 ' CHAIN ﾄ COMMON ﾉ ﾌﾟﾛｸﾞﾗﾑﾚｲ
20 ' ﾌﾟﾛｸﾞﾗﾑ B(PROB-B)
30 PRINT"ﾌﾟﾛｸﾞﾗﾑ B"
40 PRINT"ｱﾅﾀ ﾊ ";NAM$;" ｻﾝ ﾃﾞｽﾈ"
50 PRINT"ｿｼﾃ ｱﾅﾀ ﾊ ";OLD;" ｻｲ ﾃﾞｽﾈ"
60 CHAIN"0:PROG-A",10
```

13.8 ミニフロッピィディスクに関する特殊な命令

13.8.1 DSKO$ (ディスク・オー・ダラー)

任意のドライブのフロッピィディスクを，任意のトラック，セクタを指定して1レコード分のデータを記録する命令です．ただし，この命令を使用する時は，次の点に注意して下さい．

・BASICで作成されたファイルを，破壊する可能性があります．

BASICでは，ファイルをディレクトリとFATという部分で管理しています．しかし，この命令は，ユーザが任意のセクタへ1セクタ分のデータを記録できるのでこの管理を無視してデータを書込むと，BASICで作成したファイルが破壊されてしまうということです．

この命令を使用する場合は，特別な命令の扱いになります．

(1) オープン命令は，必要としない．

(2) システムランダムバッファというファイルバッファ（ファイル番号は0番です．）を使用して，FIELD# 命令でファイルバッファに割り当てる文字変数の定義をしなければなりません．

DSKO$ ドライブ番号，トラック番号，セクタ番号

- ドライブ番号：0～1まで
- トラック番号：0～39まで
- セクタ番号：1～32まで，ただし，17～32は，裏面のセクタが指定されます．

13.8.2 DSKI$ (ディスク・アイ・ダラー)

フロッピィディスクのデータを，任意のトラック，セクタを指定して，1レコード分入力する関数です．DSKI$は，DSKO$と同じように，この関数もオープン命令を使用しませんが，FIELD# 0により，システムランダムバッファに割り当てる文字変数の定義をしなければなりません．さらに，関数ですから式でなければなりません．

DUMMY$=DSKI$（ドライブ番号，トラック番号，セクタ番号）

ドライブ，トラック，セクタ番号は，DSKO＄の時と同様になります．

上（前ページ）の例のDUMMY＄は，フロッピィディスクの1レコード分のデータを，システムランダムバッファに取込むためのダミーの文字変数にすぎません．また，DUMMY＄の中には，1レコード分のデータ256バイトのうち，255バイトまでしかはいっていません．FIELD＃文で指定した文字変数を使用することによって，正しくデータを読取ることができるのです．

13.8.3 DSKF（ディスク・エフ）

フロッピィディスクの使用していないクラスタ数の値を，与えてくれる入出力関数です．また，FILESにより最後に「××× Clustres Free」と表示された値と同じ値となります．

〈プログラム例〉

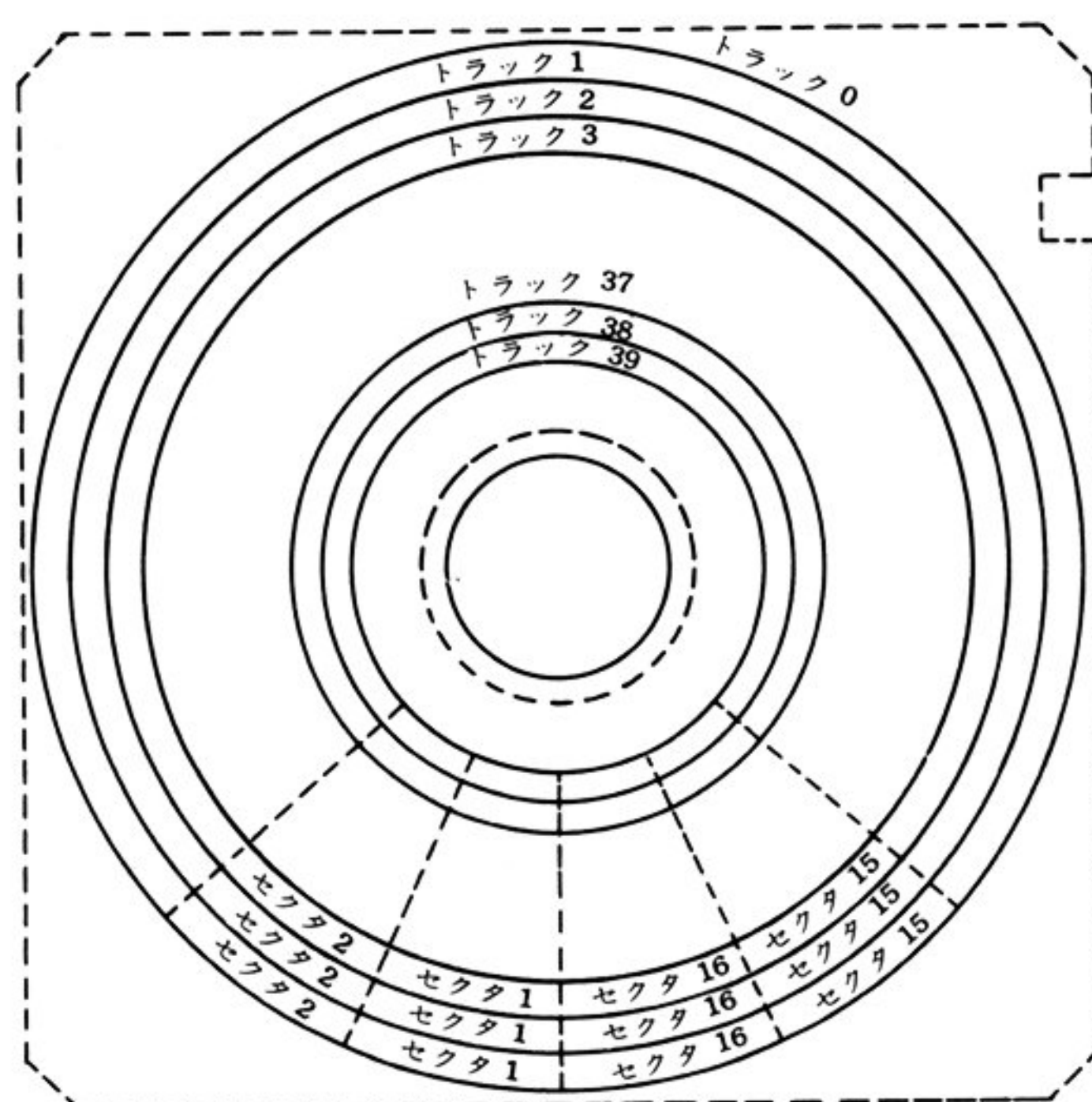
```
10 ' DSKF ｻﾝﾌﾟﾙ
20 F=DSKF(0)
30 PRINT F
40 END
RUN
 105

Ready
■
```

これは，ドライブ0のフロッピィディスクの残りクラスタをFという数値変数に入れ，このFを表示するプログラムです．

DSKF (ドライブ番号)
└── 残りクラスタ数を見たいフロッピィディスクがセットされているドライブの番号です．

13.9 ファイルの構成と管理

13.9.1 ミニフロッピィディスクの構造

FM-7において使用するミニフロッピィディスクは5.25インチ両面倍密度型であり，その構造は次のようになっています．

ミニフロッピィディスクの表をサイド0，裏をサイド1と呼び，F-BASICにおけるディスク1枚の主な仕様は，次のようになっています．

記憶容量	320 Kバイト
シリンダ数	40シリンダ
トラック数	80トラック（40トラック×2）
セクタ数	16セクタ／トラック
セクタ長	256バイト／セクタ

13.9.2 セクタアドレス

ミニフロッピィディスクのある特定のセクタを指定するには，トラック番号とセクタ番号を用いて行ないます．両面タイプの場合は，さらにサイド番号が必要ですが，F-BASICでは，サイド0，サイド1の全セクタに通し番号を付け，その通し番号とトラック番号により，セクタを指定します．物理的なセクタ番号との対応を次に示します．

	サイド0			サイド1		
セクタ	1		16	1		16
トラック 0	セクタ1	----	セクタ16	セクタ17	----	セクタ32
⋮						
トラック 39	セクタ1	----	セクタ16	セクタ17	----	セクタ32

13.9.3 クラスタ

ミニフロッピィディスクの入出力の最小単位は1セクタですが，F-BASICでは，8セクタを管理の最小単位として扱います．この単位をクラスタと呼びます．

トラック2のセクタ1〜8がクラスタ0，トラック39のセクタ25〜32がクラスタ151となります．

1セクタは256バイトですので，1クラスタは2048バイトとなります．

1クラスタ=2,048バイト=2Kバイト

13.9.4 トラックの割当て

ミニフロッピィディスクの各トラックは，次の表のように割り当てられます．

トラック番号	内容
0	IPL，ID，DISKコード
1	FAT，ディレクトリ
2〜39	ユーザ領域

・IPL（アイ・ピー・エル：Initial Program Loader）

DOSをメモリ上へ展開するためのプログラムです．

・ID（アイ・ディー：Identification）

F-BASICにおいて，使用可能なミニフロッピィディスクであることを識別する領域です．

・DISKコード（ディスク・コード）

F-BASICにおける，ミニフロッピィディスクにのみ対応する命令が格納されています．この内容は，DISKモード起動時に，FM-7本体メモリの7000（16進）番地以降へ展開されます．

・FAT（ファット：File Allocation Table）

各クラスタの使用状況を記録する領域です．

・ディレクトリ

ファイル名，ファイル形式，ファイルの位置を記録する領域です．

・ユーザ領域

ユーザ（使用者）が，ファイルを記録する領域です．

13.9.5 トラック0と1の構成

トラック0

セクタ　1　2　3　4　5　6　7　8　12　13　14　15　16

表　面　IPL　ID　リザーブ　リザーブ　DISK コード

セクタ　17　18　19　20　21　22　23　24　28　29　30　31　32

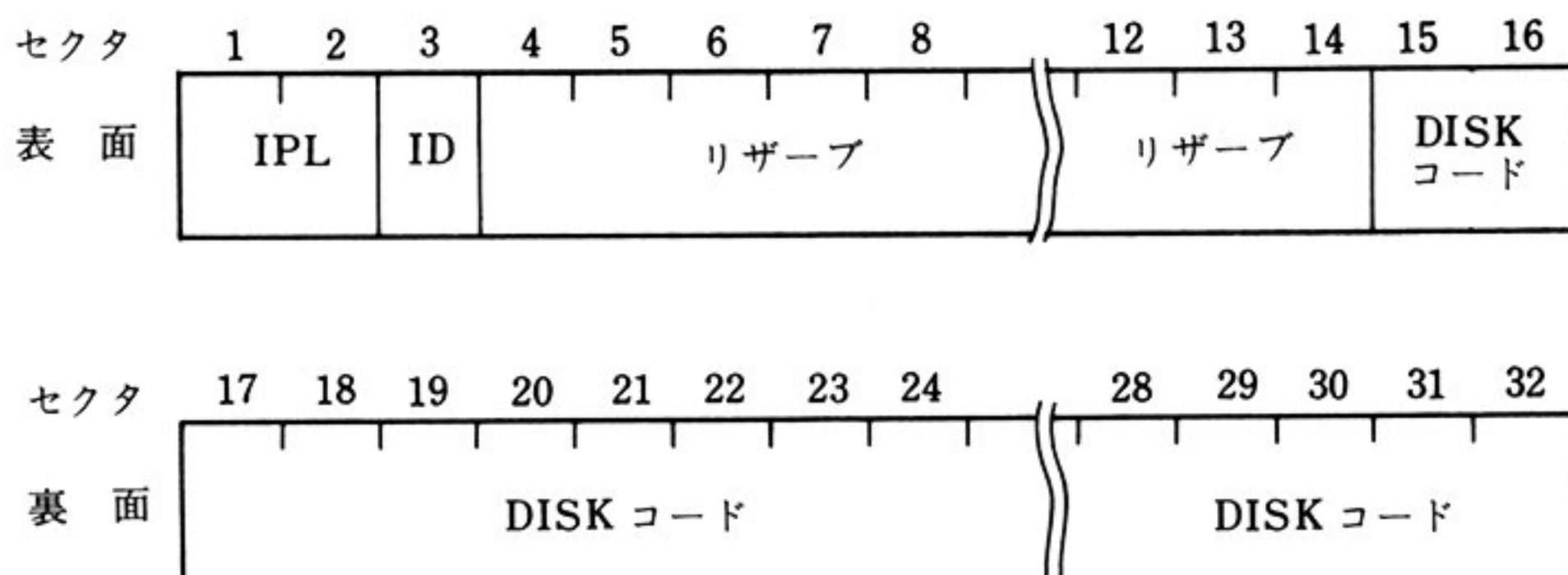
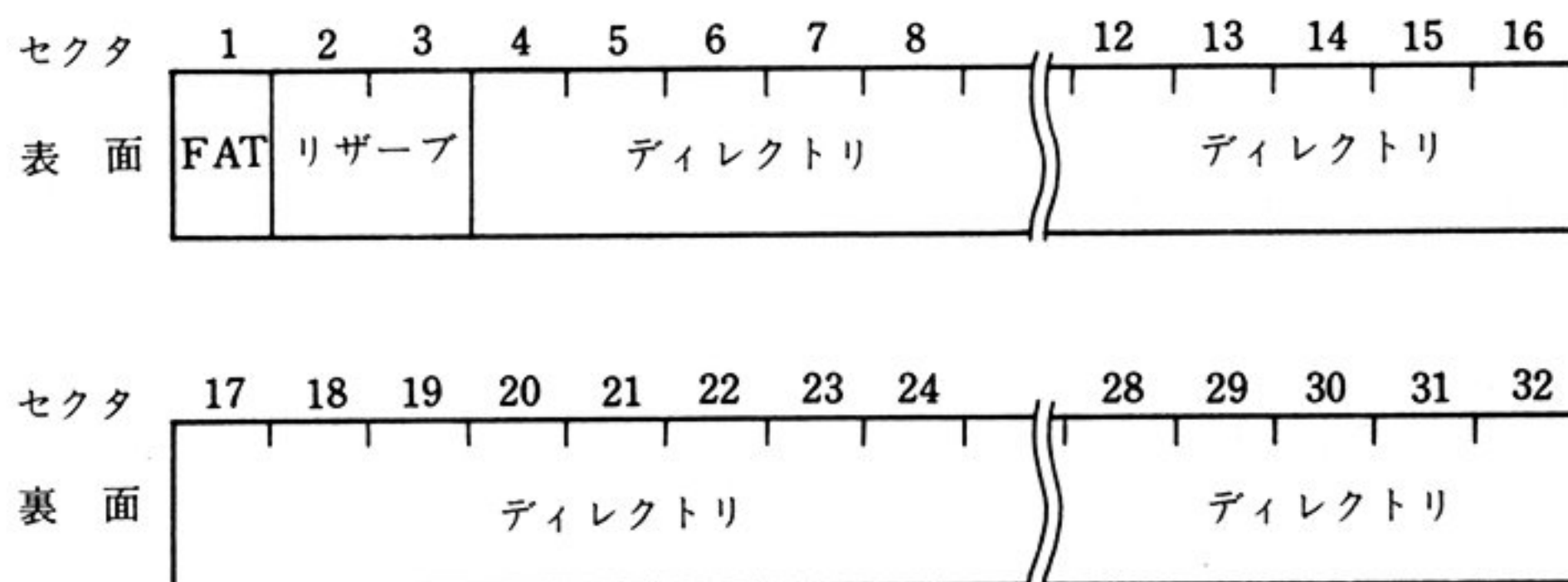
裏　面　DISK コード　DISK コード

トラック1

セクタ　1　2　3　4　5　6　7　8　12　13　14　15　16

表　面　FAT　リザーブ　ディレクトリ　ディレクトリ

セクタ　17　18　19　20　21　22　23　24　28　29　30　31　32

裏　面　ディレクトリ　ディレクトリ

13.9.6 FAT（ファット）

FATは各クラスタの使用状況を示すテーブルです．F-BASICはディスクの領域の管理を，このFATによりすべて行なっています．FATはトラック1の最初のセクタ内に書込まれ，6～157バイトがクラスタの0～151に対応して，各バイトは次の意味を持っています．

値（16進）	意　味（クラスタの使用状況）
0～97	このクラスタは使用中であり，後続するクラスタを持つ．また，その値が後続するクラスタの番号を示す．
C0～C7	このクラスタは使用中であり，ファイル最後のクラスタである．この値から16進数のBFを引いた値は，クラスタ内の実際に使用されているセクタ数を示す．
FD	クラスタ中の使用セクタはない．
FE	このクラスタはシステム領域である．
FF	このクラスタは未使用である．

13.9.7 ID（アイ・ディー）

IDとは，システム識別記号領域と呼ばれ，下記に示すシステムに関する内容を記録しています．

トラック0のセクタ3

第0～2バイト：〝S 00〞と記録されており，F-BASICにおいて使用可能であるかは，〝S〞が記録されているかにより判断されます．

第3バイト：自動スタート機能により設定されるミニフロッピィディスクのドライブ数が記録されます．

第4バイト：自動スタート機能により設定されるミニフロッピィディスクに対するファイルバッファ数が記録されます．

第5～255バイト：未使用であり，すべて〝00〞が記録されます．

13.9.8 ディレクトリ

ディレクトリは，トラック1のセクタ4～セクタ32に書込まれます．ディレクトリ部のセクタは，32バイトずつの8ブロックに分割されます．この32バイトはディレクトリスロットと呼ばれ，次の構成になっています．

ディレクトリスロット内バイト位置	内　　容
0～7	ファイル名
8～10	リザーブ用
11	ファイルタイプ 0：BASICソース 1：BASICデータ 2：機械語ファイル
12	アスキーフラグ FF：アスキー 00：バイナリ
13	ランダムアクセスフラグ 00：シーケンシャル FF：ランダム
14	ファイル先頭クラスタ番号
15～31	リザーブ用

ディレクトリスロットは232個ありますので，最大232のファイルが登録可能ですが，クラスタ数は152なので，実際登録できるファイル数は最大152となります．

13.10 ミニフロッピィディスクの破壊

ミニフロッピィディスクが急に読めなくなった……つまりエラー(Bad File Structure)が発生し，プログラムのロード・セーブ，データの読み書きができなくなるということが，使用方法，環境等により起こります.

以下に原因の説明を行ないます.

13.10.1 使用環境によるもの

(1) 低温時にミニフロッピィディスクへ記録を行ない，高温な場所においてロード・セーブを行なう. これは，ミニフロッピィディスクの温度に対する収縮によるもので，同温において使用すると，ロード・セーブ可能となります. 使用温度はなるべく急変動のないよう注意して下さい.

(2) 発電機，大きなモータ等，大きなノイズ（雑音）を発生するものの近くにおいて使用すると，クラスタ数等，FATの管理が変更される. これは，データを入力する際，ノイズにより，データが破壊されてしまうことによります. 大きなノイズを発生するものの近くでは使用しないで下さい.

(3) その他，ホコリ等によるミニフロッピィディスクへのキズ等により，読めなくなることがあります.

以上，使用環境には十分御注意下さい.

13.10.2 使用方法によるもの

(1) OPEN命令を実行後，CLOSE命令を実行しなかったために，ファイルの内容が変わってしまうことがあります.

OPEN命令を実行すると，ミニフロッピィディスクに対するバッファを，メモリ上へ設定します.

データは，このバッファを通してミニフロッピィディスクと本体との受け渡しを行ないます.

データをミニフロッピィディスクへ記録する際，バッファへデータが残っている場合は，CLOSE命令により，すべてのデータを，ミニフロッピィディスクへ記録します. このとき，CLOSE命令を実行せずに，使用中のミニフロッピィディスクを他のものと差し換え，CLOSE命令を行なうと，どうなるでしょう. 当然，差し換えた後のミニフロッピィディスクに対して，バッファのデータを記録してしまいます.

しかし，通常，ミニフロッピィディスクを差し換えた後，CLOSE命令を実行しないのに，ファイルの内容が変わってしまう場合があります. これは，〝RUN〞命令を実行したことが原因となる場合が多いのです.

プログラムを実行するための命令が〝RUN〞ですが，この命令は，プログラムの実行を行なう

前に，OPEN命令により開かれているファイルをすべて閉じる処理を行ないます．

つまり，CLOSE命令を実行したことと同じになるのです．よって，この差し換えたミニフロッピィディスクへバッファのデータを記録し，内容が変更されてしまいます．

以上のことより，OPEN命令により開かれたファイルは，同一プログラム上において，必ずCLOSE命令により閉じるようにして下さい．

(2) 読取り，または記録中に，ミニフロッピィディスクを抜き取ったとき．

ミニフロッピィディスクに対して，データの記録や読取りは，すべてヘッドを介して行ないます．

また，ミニフロッピィディスクユニットのドライブカバーは，ロックされませんので，いつでもミニフロッピィディスクを取り出すことができます．

このことより，ミニフロッピィディスクへの記録や読取りの最中にドライブカバーを開け，ミニフロッピィディスクを抜き取った場合，ヘッドにより，キズを付けたり，データが正しく記録されていないといったことになりますので，ミニフロッピィディスクの抜き取りには，十分御注意下さい．

ディスプレイ上へ〝Ready〞が表示されても，ミニフロッピィディスクに対しての処理は終了していないといったことがありますので，〝Ready〞が表示され，4～5秒待ち，ミニフロッピィディスクユニットより〝カチャッ〞というヘッドがミニフロッピィディスクより離れる音が聞こえた後，抜き取りを行なって下さい．

13.11 ディスクユーティリティ

F-BASICのシステムディスクには，DISKコードと次のプログラムが記録されています．

(1) DFMCD ………SYSDSK補助プログラム

(2) MCOPY ………VOLCOPY補助プログラム

(3) SYSDSK………システムディスク作成用プログラム

(4) VOLCOPY……ディスクのコピーを行なうプログラム

(5) AUTOUTY……自動スタートユーティリティ

(6) PFDEF ………PFキーの設定ユーティリティ

(7) DEMO ………デモ用プログラム

DFMCD，MCOPYは機械語プログラムで，他はBASICのソースプログラムです．

以上のプログラムファイルにより，実行する目的を示します．

(1) システムディスクの作成

ミニフロッピィディスクをFM-7において使用可能とする．

SYSDSK, DFMCDを使用します．

(2) ミニフロッピィディスクのコピー

ミニフロッピィディスクを，すべて別のミニフロッピィディスクへ複写する．

VOLCOPY，MCOPY

(3) プログラムの自動スタート

プログラムを，F-BASIC V3.0 DISKモード起動時に，自動スタートさせる．

AUTOUTY

(4) PFキーの初期設定

F-BASIC V 3.0 DISKモード起動時のPFキー(プログラマブル・ファンクション・キー)を設定する．

PFDEF

次に，各プログラムの使用方法を示します．

ユーティリティにおいて，ミニフロッピィディスクの1ドライブのものと2ドライブのものを使用することにより，使用方法が異なりますので，それぞれを分けて説明します．

13.11.1 システムディスクの作成

プログラムは〝SYSDSK(シスディスク)″を実行します．このプログラムは，F-BASICのシステムディスクを作成するためのプログラムで，BASICで記述されています．

SYSDSKプログラムは，ディスクのフォーマッティングとDISKコードのコピーを行ないます．

市販されているディスクは，そのままではデータやプログラムを読み書きできません．ディスクを使用できるようにするには，ディスクの初期化を行なうことが必要です．新品のディスクを，F-BASIC V 3.0 DISKモードで使用できるようにするためには，次の手続きが必要です．

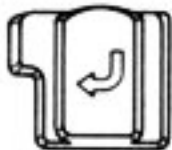
(1) ミニフロッピィディスクユニットの1ドライブを使用する場合

市販のミニフロッピィディスクは，管理情報等が書かれていないため，そのままではデータやプログラムの読み書きができません．そのためディスクのフォーマッティングを行ない，読み書き可能な状態にします．その後，DISKコードのコピーを行ないます．

「FACOM Floppy Disk 2D」と記入されているミニフロッピィディスクの場合は，フォーマッティングずみですので，DISKコードのコピーだけを行なって下さい．

以下に1ドライブ使用時のSYSDSKプログラムの実行手順を並べます.

① システムディスクをドライブ0に挿入して

RUN "SYSDSK" ⏎

を入力します.

② 画面に

***FLOPPY DISK FORMATTING ***

DISK FORMATTING?

と表示されます. フォーマッティングを行なう場合は, Y ⏎ を入力します.

その他のものが入力されると, フォーマッティングは実行されず, ⑥のDISKコードのコピーを行なう処理へ分岐します.

「FACOM Floppy Disk 2D」のミニフロッピィディスクを使用する場合は, , N ⏎ を入力します.

③ ②の操作でYを入力すると, 画面に

WHAT DISK DRIVE ?

と表示されます. フォーマッティングするミニフロッピィディスクを挿入するドライブ番号を入力します. 1ドライブを使用しますので, ドライブ番号は0となります.

0 ⏎ を入力します.

④ ドライブ番号を入力すると, 画面に

SET THE FORMATTING DISK ON DRIVE 0

READY ?

と表示されます.

システムディスクを取り出した後, フォーマッティングを実行するミニフロッピィディスクを挿入して, Y ⏎ と入力して下さい. その他のものが入力された時は, ③にもどります.

⑤ 約40秒ほどでフォーマッティングが終了し, 画面に

DISK FORMATTING COMPLETED !!

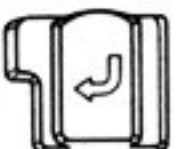
と表示されます.

(注) もし, フォーマッティング作業中に, ミニフロッピィディスクにエラーがあった場合は,

DISK I/O ERROR

と表示され, SYSDSKプログラムは終了します.

⑥ すでに, フォーマッティングの終了しているミニフロッピィディスクに対して, DISKコードのコピーを行ないます.

画面に, 次のように表示されます.

```
*** DISK BASIC SYSTEM COPY ***
1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=
```

1は，ドライブ0からドライブ1にシステムコードをコピーします．2は，ドライブ0からドライブ0にシステムコードをコピーします．2⏎と入力して下さい．

画面に　READY？　と表示されます．ミニフロッピィディスクユニットより，フォーマッティングずみのミニフロッピィディスクを取り出し，システムディスクを再び挿入した後に，Y⏎を入力して下さい．

⑦ システムディスクからDISKコードが，メモリ上に読込まれた後，画面に

```
SET THE DESTINATION DISK
READY ?
```

と表示されます．システムディスクを取り出し，フォーマッティングずみのミニフロッピィディスクを挿入した後，Y⏎を入力します．

メモリ上のDISKコードのコピー終了後

```
SYSTEM CODE COPY COMPLETED !!
```

と表示され，SYSDSKプログラムの実行は終了します．

⑧ SYSDSKプログラムが終了すると，BASICコマンドレベルにもどりますので，DSKINIコマンドを実行します．

DSKINIの後に，新しく作成しようとするミニフロッピィディスクの挿入されているドライブの番号0番を指定します．

DSKINI　0⏎を入力します．

画面に

```
DSKINI 0
Are you sure(Y or N)?
```

と表示されますので,フォーマッティングを行なったミニフロッピィディスクが挿入されているのを確認後，⏎を入力します．

DSKINIコマンドを実行すると，ディレクトリの初期化が行なわれます．

以上で，システムディスク作成の手順を終了します．

実 行 例

```
RUN "SYSDSK"

*** FLOPPY DISK FORMATTING ***

DISK FORMATTING ? Y
WHAT DISK DRIVE ? 0
SET THE FORMATTING DISK ON DRIVE 0
READY ? Y
DISK FORMATTING COMPLETED !!
*** DISK BASIC SYSTEM COPY ***

1. DRIVE 0 -->1
2. DRIVE 0 -->0
= 2
READY ? Y
SET THE DESTINATION DISK
READY ? Y
SYSTEM CODE COPY COMPLETED !!

Ready
DSKINI 0
Are you sure(Y or N)? Y

Ready
```

(2) 2ドライブのミニフロッピィディスクユニットを使用した場合

2ドライブのミニフロッピィディスクユニットを使用した場合の使用方法を以下に示します.

[第1ステップ]

システムディスクに格納されているSYSDSKプログラムで,新品のディスクに,決められたデータを書込みます.これにより,ディスクはリード/ライト可能となりますが,まだF-BASIC V3.0 DISKモードでは,このディスクを使用できません.

[第2ステップ]

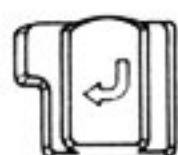
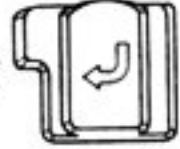
第1ステップ後,SYSDSKプログラムを引続き実行し,DISKコードのコピーを行ないます.

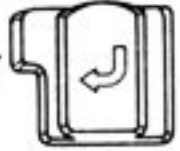
DSKINI命令で,ディレクトリの内容を初期化します.これで,新品のディスクがF-BASIC V3.0 DISKモードのディスクとして使用できるようになります.

システムディスク作成の具体的手順

[第1ステップ]

① システムディスクをドライブ0に入れます.

RUN "SYSDSK" ↵

を入力すると，システムディスクに格納されているSYSDSKプログラムが起動されます．

② 画面に

　　　＊＊＊ FLOPPY DISK FORMATTING ＊＊＊

と，メッセージが表示されます．

次に，DISK FORMATTING ? と表示され，フォーマッティングを行なうかどうかを聞いてきますので，フォーマッティングをする場合は，Yと を入力します．その他のものが入力されたときは，フォーマッティングは実行されず，ディスクのシステムコピーのところに分岐します．(FACOM Floppy Disk 2Dを使用のときはNと を入力してください．)

Yと を入力すると，WHAT DISK DRIVE ? と聞いてきますので，フォーマッティングするディスクを挿入するドライブ番号1番と， を入力します．そうすると，自動的にDFMCDの機械語プログラムがロードされ，次に，SET THE FORMATTING DISK ON DRIVE 1というように表示され，READY ? と聞いてきます．

Y と入力すると，フォーマッティングが開始され，その他のものが入力されたときには，再びWHAT DISK DRIVE ? のメッセージ出力のところにもどります．

フォーマッティングが開始されてから約40秒間，ドライブが動いてフォーマッティング作業をした後，DISK FORMATTING COMPLETED !! の表示がされ，フォーマットは終了します．

フォーマッティング作業中にディスクにエラーのあった場合は，DISK I/O ERROR と表示され，SYSDSKプログラムは終了します．

[第2ステップ]

第1ステップにおいて，フォーマッティングが正常に終了した場合，または，第1ステップでフォーマッティングを指示しなかった場合には，第2ステップが起動されます．

① 画面に

```
*** DISK BASIC SYSTEM COPY ***
1. DRIVE 0 --> 1
2. DRIVE 0 --> 0
=
```

と表示されますので，番号の1を指定します．1の指定をすると，ドライブ0からドライブ1にシステムコードがコピーされます．

1を入力すると，READY ? と聞いてきますので，ドライブ1にフォーマッティング済みのディスクを挿入して，Y と入力します．

1の指定の場合は自動的にコピーが行なわれ，

　　　SYSTEM CODE COPY COMPLETED !!

と表示されます．

[第3ステップ]

SYSDSKプログラムが終了すると，BASICのコマンドレベルにもどり，READYが表示され，コマンド待ちとなりますので，

DSKINI 1 ⏎

を実行します．新しく作成しようとするディスクが挿入されている，ドライブの番号を指定します．

画面に Are you sure (Y or N) ? と表示されますので，実行してよいときはYを，やめたいときはNを入力します．Yを入力すると，ディレクトリの初期化が行なわれます．

実 行 例

```
RUN "SYSDSK"

*** FLOPPY DISK FORMATTING ***

DISK FORMATTING ? Y
WHAT DISK DRIVE ? 1
SET THE FORMATTING DISK ON DRIVE 1
READY ? Y
DISK FORMATTING COMPLETED !!
*** DISK BASIC SYSTEM COPY ***

1. DRIVE 0 -->1
2. DRIVE 0 -->0
= 1
READY ? Y
SYSTEM CODE COPY COMPLETED !!

Ready
DSKINI 1
Are you sure(Y or N) ? Y

Ready
```

13.11.2 ミニフロッピィディスクのコピー

プログラムは，"VOLCOPY（ボルコピー）"を使用します．

このプログラムは，1枚のディスクの内容を完全に他のディスクにコピーします．RUN "VOLCOPY" とすると，プログラムがメモリ上にローディングされ，実行が開始されます．

(1) ミニフロッピィディスクユニットの1ドライブを使用する場合

以下に1ドライブ使用時のVOLCOPYプログラムの実行手順を述べます．

① VOLCOPY プログラムの入ったミニフロッピィディスクを挿入して，

RUN "VOLCOPY" ⏎

と入力します．

② 画面に

```
*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
```

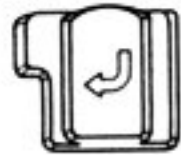
と表示されます．

1は，ドライブ0番の内容が，ドライブ1番へ，

2は，ドライブ1番の内容が，ドライブ0番へ，

3は，ドライブ0番の内容が，ドライブ0番へ，

それぞれコピーされます．

1ドライブを使用しますので3の，0番から0番へのコピーを指定します．

3 ⏎ を入力します．

③ 画面に READY ? が表示されますので，

ソースディスク（コピーの元になるディスク）を，挿入した後，Y ⏎ を入力します．

その他のものを入力すると，②の入力待ちの状態になります．

④ ドライブ0のミニフロッピィディスクのトラック0と1の内容が，メモリに読込まれた後，画面に

```
TRACK = 0 1
SET THE DESTINATION DISK !!
READY ?
```

と表示されます．新しくコピーされるミニフロッピィディスクを挿入した後，Y ⏎ を入力します．

⑤ 画面に

```
SET THE SOURCE DISK !
READY ?
```

と表示されますので，再度ソースディスクをドライブ0に挿入して，Y ⏎ を入力します．

⑥ 画面に

```
TRACK = 2 3
SET THE DESTINATION DISK !!
READY ?
```

と表示されますので，同様に，4），5）の手順を〝COPY FINISHED !″と表示されるまで繰り返して下さい．

実 行 例

```
Ready
RUN "VOLCOPY"
*** DISK COPY ***
1.  DRIVE 0 --> 1
2.  DRIVE 1 --> 0
3.  DRIVE 0 --> 0
=   3
READY ? Y
TRACK =  0  1
SET THE DESTINATION DISK !!
READY ? Y
SET THE SOURCE DISK !!
READY ? Y
TRACK =  2  3
SET THE DESTINATION DISK !!
READY ? Y
SET THE SOURCE DISK !!
READY ? Y
TRACK =  4  5
            :
            :
            :
            :
            :
SET THE SOURCE DISK !!
READY ? Y
TRACK =  34  35
SET THE DESTINATION DISK !!
READ ? Y
SET THE SOURCE DISK !!
READY ? Y
TRACK =  36  37
SET THE DESTINATION DISK !!
READY ? Y
SET THE SOURCE DISK !!
READY ? Y
TRACK =  38  39
SET THE DESTINATION DISK !!
COPY FINISHED !
Ready
■
```

(2) ミニフロッピィディスクユニットの2ドライブを使用する場合

ファイル名が"VOLCOPY"のプログラムを実行します.

```
RUN "VOLCOPY" ⏎
```

実行後画面に,

```
*** DISK COPY ***
1. DRIVE 0 -->1
2. DRIVE 1 -->0
3. DRIVE 0 -->0

=
```

と表示されますので, 1を入力します. 1を入力すると, ドライブの0番から1番へコピーされます. 2番を指定するときは, システムディスクが破壊されないように, 注意をしなければなりません.

1を入力すると, MCOPYが自動的にメモリ上にロードされ, READY ? と表示されます.

1の指定の場合は, ドライブ1に新しくコピーするディスクを挿入し, Y⏎を入力します. 処理の進行とともにトラック番号が表示されます. すべての処理が正常に終わると,

```
COPY FINISHED !
```

と表示され, 処理の途中でエラーのあったときは,

```
COPY PROGRAM ABORTING
```

と表示され, プログラムは終了します.

実 行 例

```
Ready
RUN "VOLCOPY"
*** DISK COPY ***
1. DRIVE 0 --> 1
2. DRIVE 1 --> 0
3. DRIVE 0 --> 0
=  1
READY ? Y
TRACK =  0   1
TRACK =  2   3
TRACK =  4   5
TRACK =  6   7
         :
         :
         :
TRACK =  34  35
TRACK =  36  37
TRACK =  38  39
COPY FINISHED!

Ready
```

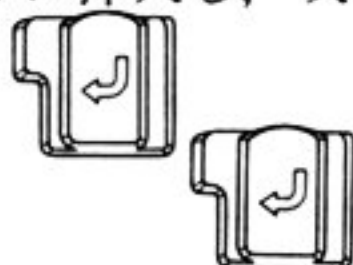
13.11.3 補助プログラム (DFMCD/MCOPY)

DFMCDは，SYSDSKプログラム実行時に，自動的に呼出されて実行されます．またMCOPYは，VOLCOPYプログラム実行時に，自動的に呼出されて実行されます．したがって，SYSDSK，MCOPYプログラムを実行するためには，これらのプログラムの入っているディスクを，ドライブユニット0に挿入しておく必要があります．

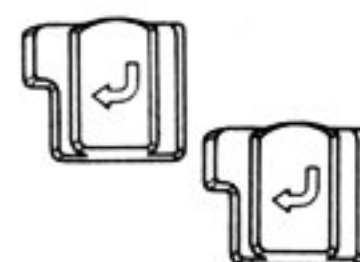
DFMCD，MCOPYのコピーを作成するためには，ユーティリティのVOLCOPYを使用するか，次の手順を行なうことが必要です．(　　) 内はMCOPYの場合です．

(1) システムディスクをドライブ0に挿入し，次のようにロードコマンドを実行します．

LOADM "DFMCD" ⏎

LOADM "MCOPY" ⏎

(2) (1)によりプログラムがメモリ上に読込まれたので，セーブコマンドを実行し，記録するためのディスクに書込みます．

SAVEM "1：DFMCD", &H5000, &H512F, &H5000 ⏎

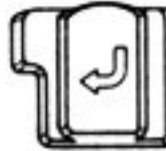
(SAVEM "1：MCOPY", &H2000, &H20E8, &H201A) ⏎

これでDFMCD (MCOPY) のコピーは終了します．SAVEMを実行するとき，ファイル名や，各番地を変更すると，SYSDSK，VOLCOPYプログラムは正常に動作しなくなります．

なお，上の例では，1番のディスクユニットに書込みましたが，0番のユニットに書込むこともできます．そのときは，"0：DFMCD" あるいは "0：MCOPY" とします．

13.11.4 プログラムの自動スタート

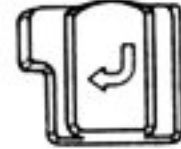
プログラムは，"AUTOUTY (オートユーティーワイ)" を実行します．

このユーティリティは，ユーザプログラムを電源投入とともに，"STARTUP" というファイル名を持つプログラムファイルを実行させるための準備を行なうユーティリティです．(STARTUPと連結されて，ユーザプログラムは起動されるようになっています．)

通常システムを起動させますと，ディスプレイ画面には，ミニフロッピィディスクユニットのドライブを使用する数の質問とフロッピィディスクに対するファイルバッファ数を必ず聞いてきます．この入力を行なわないと，F-BASIC V3.0 DISKモードは起動されません．

そこで，この入力をあらかじめシステムディスクに記録しておくことにより，入力を省略できるようにします．また，STARTUPに起動させるためのプログラムを，RUN "ファイル名" とプログラムしているとDISKモードがシステム起動された後に，プログラムがロードされ実行させるようになっています．

以上がオートスタートユーティリティのしくみです．

AUTOUTYは，システム起動に際しての初期設定を，システムディスクへ記録したり，また，STARTUPを起動時において実行するユーティリティです．また，STARTUPは，起動したユーザプログラムを指定するためのプログラムファイルです．

オートスタート ユーティリティの操作説明

(1) システムディスクをドライブ0にセットして，RUN "AUTOUTY" [↵] と入力し，オートスタートユーティリティプログラムを起動して下さい．

(2) オートスタートユーティリティの起動メッセージ

　　*** AUTO START UTILITY ***

(3) HOW MANY DISK DRIVES (0−4) ?

使用するミニフロッピィディスクのドライブ数を0～4の間の数字で入力し，[↵] を押して下さい．0の入力は，自動スタートを行なわないことを示します．

(4) HOW MANY DISK FILES (0−15) ?

プログラムで使用したいファイルバッファの数を0～15の間の数字で入力し，[↵] を押して下さい．

(5) READY ?

すべてOKですかと聞いてきますのでOKなら [Y] と入力して下さい．ここで，必ずシステムディスクがドライブ0に入っていることを確認して下さい．

(6) 以上，入力を行なった内容をフロッピィディスクに記録し終わると次の表示を行ないます．

　　COMPLETED!!

システムディスク以外のミニフロッピィディスクが入っていた場合，以下に示す表示を出力して，プログラムは終了します．

　　DISK ON DRIVE 0 IS NOT SYSTEM !!

STARTUPは，通常のBASICプログラムファイルです．ユーザが起動させたいプログラムをSTARTUPというプログラムファイルとして作成します．

例 AUTOUTYとSTARTUP

```
RUN "AUTOUTY"

*** AUTO START UTILITY ***

HOW MANY DISK DRIVES(0-4)? [↵]
HOW MANY DISK FILES(0-15)? [↵]
SET THE SYSTEM DISK ON DRIVE 0
READY ? Y [↵]
COMPLETED !!

Ready
```

オートスタート機能により最初に起動されるプログラムは『STARTUP』というプログラムです.

STARTUPのプログラム例を示します.

```
10 '  POWER ON START PROGRAM
20 '  SET THE PROGRAM NAME
30 COLOR4:PRINT "**** START PROGRAM IS DEMO ****"
40 RUN "DEMO"
```

13.11.5 PFキーの初期設定

F-BASIC V3.0 DISKモード起動時において, 10個のPFキー（プログラマブル・ファンクション・キー）の内容は, システムディスクに記録されたものが設定されます.

この, システムディスクのPFキーに対する内容を変更することにより, DISKモード起動時におけるPFキーの内容が変更できます.

ファイル名 "PFDEF" のプログラムは, このPFキーに対するシステムディスクの内容変更を, ディスプレイ上によって行なうユーティリティです.

以下に使用方法を示します.

(1) F-BASIC V3.0のシステムディスクをミニフロッピィディスクユニットのドライブ0へ挿入し, "PFDEF" プログラムを実行します.

```
RUN "PFDEF" ⏎
```

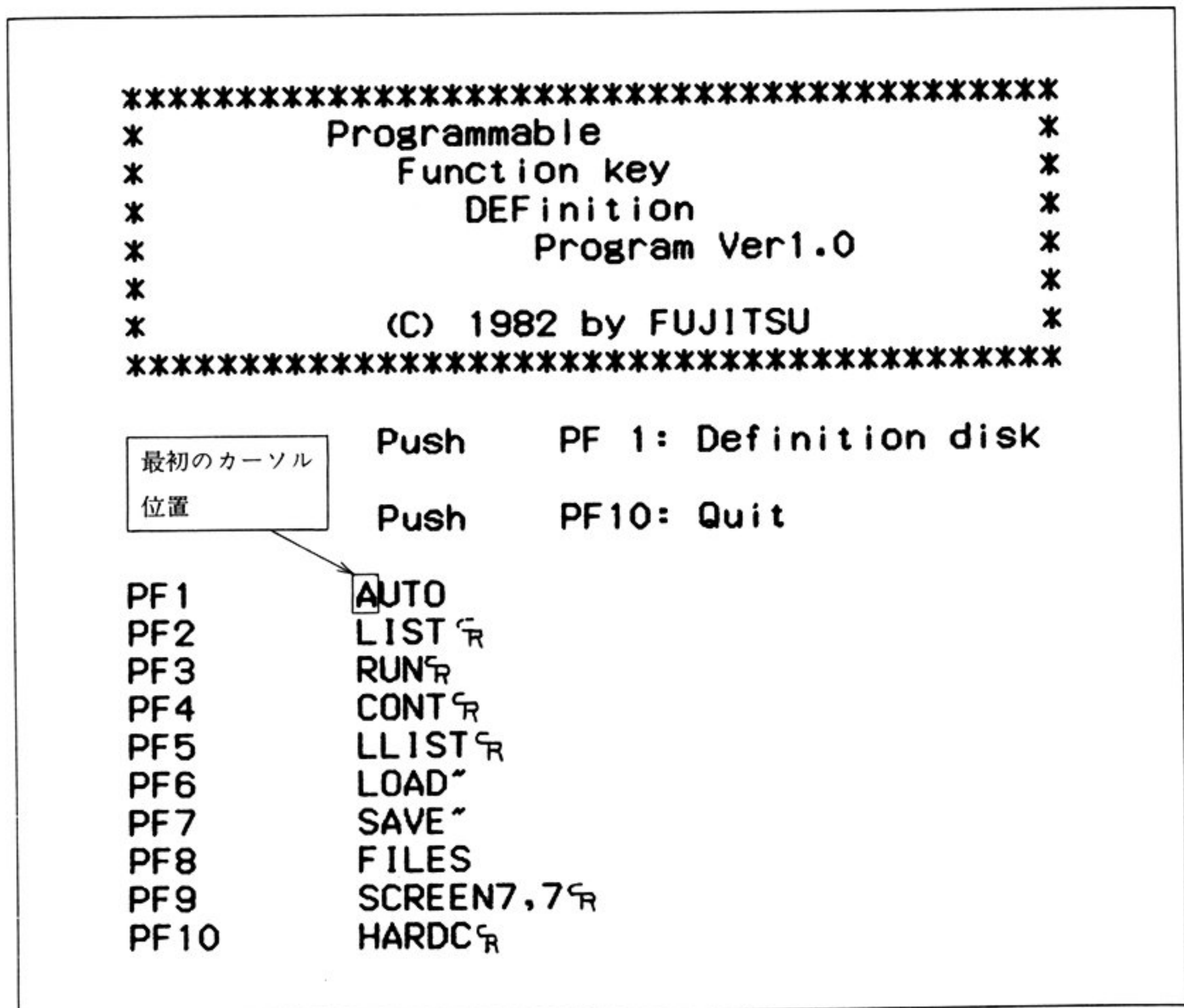
次のように画面表示されます.

```
*******************************************
*        Programmable                     *
*           Function key                  *
*              DEFinition                 *
*                 Program Ver1.0          *
*                                         *
*          (C) 1982 by FUJITSU            *
*******************************************

          Insert disk drive 0

              Ready (Y/N)?
```

ここで，F-BASIC V3.0 のシステムディスクが挿入されていれば，Y ⏎ と入力します．

(2) 次に，画面へ現在設定されている PF キーの内容とカーソルが，ディスプレイへ表示されます．

```
****************************************
*          Programmable                *
*              Function key            *
*                  DEFinition          *
*                      Program Ver1.0  *
*                                      *
*          (C) 1982 by FUJITSU         *
****************************************

            Push      PF  1: Definition disk

            Push      PF10: Quit

PF1         AUTO
PF2         LIST⏎
PF3         RUN⏎
PF4         CONT⏎
PF5         LLIST⏎
PF6         LOAD"
PF7         SAVE"
PF8         FILES
PF9         SCREEN7,7⏎
PF10        HARDC⏎
```

最初のカーソル位置

上に示す画面において，カーソル移動キー（↑ ↓ ← →）によりカーソルを，変更する PF キーの内容を表示しているところへ移動させ，変更する内容を入力します．

変更の終了は PF1 を入力して下さい．PF1 入力により，変更内容をシステムディスクへ記録します．記録が終了すると，再度(1)の画面へもどります．

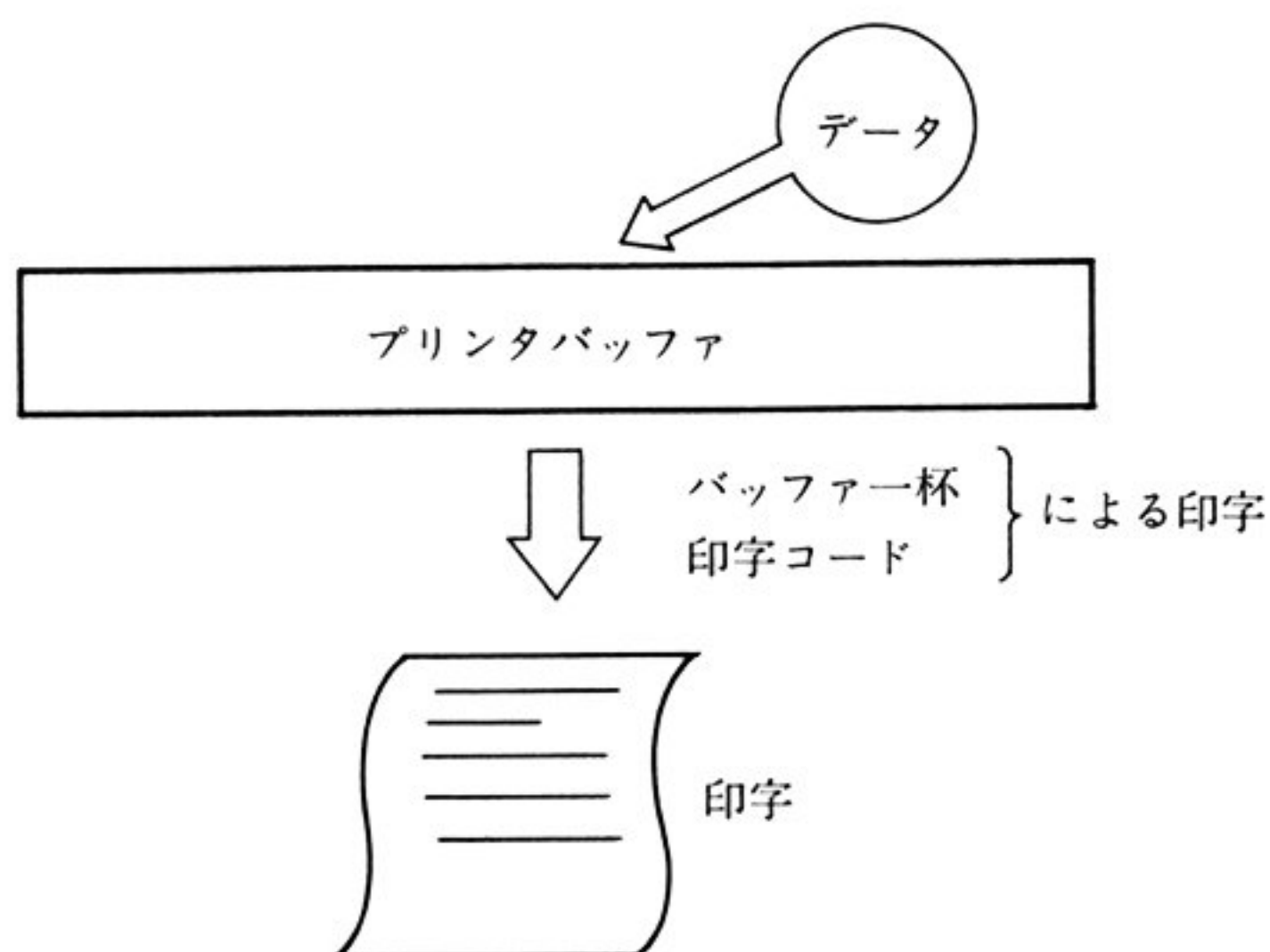
処理を終了する場合は，ここで N ⏎ と入力して下さい．

再度実行するか，または，入力した内容の確認は，Y ⏎ と入力して下さい．この画面を表示中においての処理終了は，PF10 キーを入力して下さい．

第14章 プリンタ

14.1 バッファとカウンタ

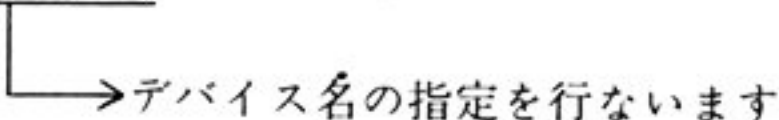
プリンタにおいても，ミニフロッピィディスクと同様な“バッファ”があります．ミニフロッピィディスクの場合，バッファは，メモリ上に設定されましたが，プリンタの場合は，プリンタ内部にバッファを持っています．バッファの大きさは，機種により，80バイトまたは136バイト分と異なり，1行分のデータを持っています．

FM-7からのデータは，1つ1つプリンタのバッファへ送出され，バッファが一杯になるか，または印字コードを送出することにより印字を行ないます．

また，バッファへのデータ数のカウントを，F-BASICにより行なっており，バッファが一杯になったことも，このカウンタによって認識することができるのです．

このカウンタは，同一行においてのデータ数をカウントしており，改行されることにより，クリアされます．

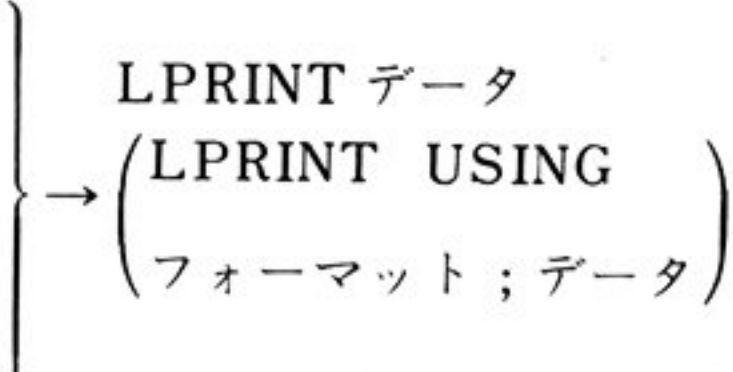
14.2 F-BASICにおけるプリンタに対する命令

以下に，F-BASICにおける，プリンタに対する命令を示します．

種　別	命　　令	処　理　内　容
コマンド	LIST LLIST	メモリ内にあるプログラムの全部または，一部をプリンタへ出力する．
	HARDC	画面をプリンタへハードコピーする．
入出力ステートメント	PRINT# LPRINT	データをプリンタへ出力する．
	PRINT# USING LPRINT USING	データを指定した書式により，プリンタへ出力する．
	OPEN	プリンタに対するファイルを開く．
	CLOSE	プリンタに対するファイルを閉じる．
一般関数	POS LPOS	プリンタヘッドの水平位置を与える．
	TAB	指定した位置までプリンタヘッドを移動させる．

上の表において，同じわく内へ記している命令は，同じ機能を持ちますが，扱いが異なります．

これは，他のデバイスと共用できる命令であるため，その命令において，デバイス名をプリンタと指定を行なうことが必要です．

わく内の上段に記した命令が，他のデバイスと共用できるものであり，下段に記した命令はプリンタのみ対応します．

以下に扱い方の違いを示し，扱う内容の説明は，プリンタにのみ対応する命令により行ないます．

（1） コマンド

LIST "LPT0:" →LLIST

（"LPT0:" →デバイス名の指定を行ないます）

デバイス名の指定が不要です．

（2） 入出力ステートメント

```
OPEN "O", #1, "LPT0:"
PRINT #1, データ
(PRINT #1 USING フォーマット ; データ)
CLOSE
```

→ LPRINT データ
（LPRINT USING フォーマット ; データ）

ファイルを開き，ファイル番号を指定し，ファイルを閉じるといった処理が省略されます．

(3) 一 般 関 数

OPEN "O", #1, "LPT0:"
A=POS(1)　　　　　　　　　→ LPOS(1)
CLOSE

入出力ステートメント同様，ファイルに対する処理を省略できます．

14.3 プログラムリストの出力(LLIST)

メモリ内のプログラムの内容を，プリンタへ全部または一部出力します．

LLIST [行番号1] [{,-} [行番号2]]

〔説 明〕

・行番号を指定しないとき，およびコンマまたはハイフォンのみ指定した場合は，プログラムの全てを出力します．

・行番号1のみ指定した場合は，指定した行を出力します．

・行番号1とコンマまたはハイフォンを指定すると，行番号1から最後の行までを出力します．

・コンマまたはハイフォンと行番号2を指定すると，プログラムの最初の行から行番号2までを出力します．

・両方の行番号を指定すると，その間の全行内容を出力します．

・プログラムの実行中にエラーが起き，コマンド受付状態にもどったときは，行番号の代わりにピリオド(・)のみを指定すると，エラーの起きた行の内容を出力します．

・省略形は使用できません．

LLIST文を使用して出力した例

以下のプログラムが，メモリ上にあるとき，LLIST文を使用して，プログラムリストを出力した例を示します．

```
10 CLS:'    ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
20 FOR Y=25 TO 200 STEP20
30 FOR X=40 TO 640 STEP70
40 HANKEI=10
50 FOR I=1 TO 7
60 IRO=INT(RND(1)*6)+1
70 CIRCLE (X,Y),HANKEI,IRO,.4495,,,,OR
80 HANKEI=HANKEI+5
90 NEXT I,X,Y
100 END
```

＜使用例1＞

LLIST　　　または {LLIST , / LLIST -}

```
10 CLS:'    ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
20 FOR Y=25 TO 200 STEP20
30 FOR X=40 TO 640 STEP70
40 HANKEI=10
50 FOR I=1 TO 7
60 IRO=INT(RND(1)*6)+1
70 CIRCLE (X,Y),HANKEI,IRO,.4495,,,,OR
80 HANKEI=HANKEI+5
90 NEXT I,X,Y
100 END
```

＜使用例2＞

LLIST 30

```
30 FOR X=40 TO 640 STEP70
```

＜使用例3＞

LLIST 60,　　　または LLIST 60－

```
60 IRO=INT(RND(1)*6)+1
70 CIRCLE (X,Y),HANKEI,IRO,.4495,,,,OR
80 HANKEI=HANKEI+5
90 NEXT I,X,Y
100 END
```

＜使用例4＞

LLIST, 20　　　または LLIST－20

```
10 CLS:'    ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
20 FOR Y=25 TO 200 STEP20
```

＜使用例5＞

LLIST 40, 70　　　または LLIST 40－70

```
40 HANKEI=10
50 FOR I=1 TO 7
60 IRO=INT(RND(1)*6)+1
70 CIRCLE (X,Y),HANKEI,IRO,.4495,,,,OR
```

14.4 プリンタへのデータ出力

プリンタへデータを出力する方法は，以下の2通りです．

(1) OPEN命令によりプリンタに対するファイルを開き，そのファイルに対してデータの出力を行なう．

(2) プリンタに対するデータ出力命令を用い，出力する．データ出力命令とは，LPRINT，LLIST等のことです．

以下に2種類の出力方法について説明します．

(1) OPEN命令による方法

```
10 OPEN "O", #1, "LPT0:"
```

- "LPT0:"→プリンタに対するデバイス名を指定します．
- #1→プリンタに対するファイルのファイル番号を指定します．ファイル番号は，＃1～16まで指定できます．
- "O"→OUTPUT：出力を表します．

```
20 PRINT #1, "AAA"
```

- "AAA"→出力するデータです．
- #1→出力するファイルに対応するファイル番号を指定します．
- PRINT→他のデバイスと共用できる命令です．

```
100 CLOSE
```

- CLOSE→OPEN命令により，開いたファイルは，CLOSE命令により必ず閉じます．

上に示した方法は，PRINT＃命令ですが，他の命令も同様に扱います．

(2) プリンタに対する命令を用いた場合

プリンタに対する命令を用いる場合は，OPEN命令により，プリンタに対するファイルを開く必要はありません．

命令を実行することにより，その命令がOPEN命令とCLOSE命令やデータ出力の処理を行ないます．

(例)

LPRINT "AAA" [↵]

上に示す例の入力により，"AAA"というデータがプリンタへ出力されます．

このように，OPEN命令を使用する必要もないことより，多くの場合，プリンタに対する命令は，LPRINT，LLIST等の命令を使用します．

以下に，プリンタに対する出力例を示します．

プログラム例として，ボウリングのスコアを画面へ表示して，その後プリンタへ出力し，最後に画面の表示をハードコピーするプログラムを示します．

ハードコピーとは，画面に表示されているデータを，画面のイメージによりプリンタへ出力します．ハードコピーするためには，“HARDC（ハードコピー）”命令を使用します．

プログラム例の画面へ出力する命令と，プリンタへ出力する命令が似ていることに注意して下さい．出力する形式，データの内容は，まったく同じものを使用しています．

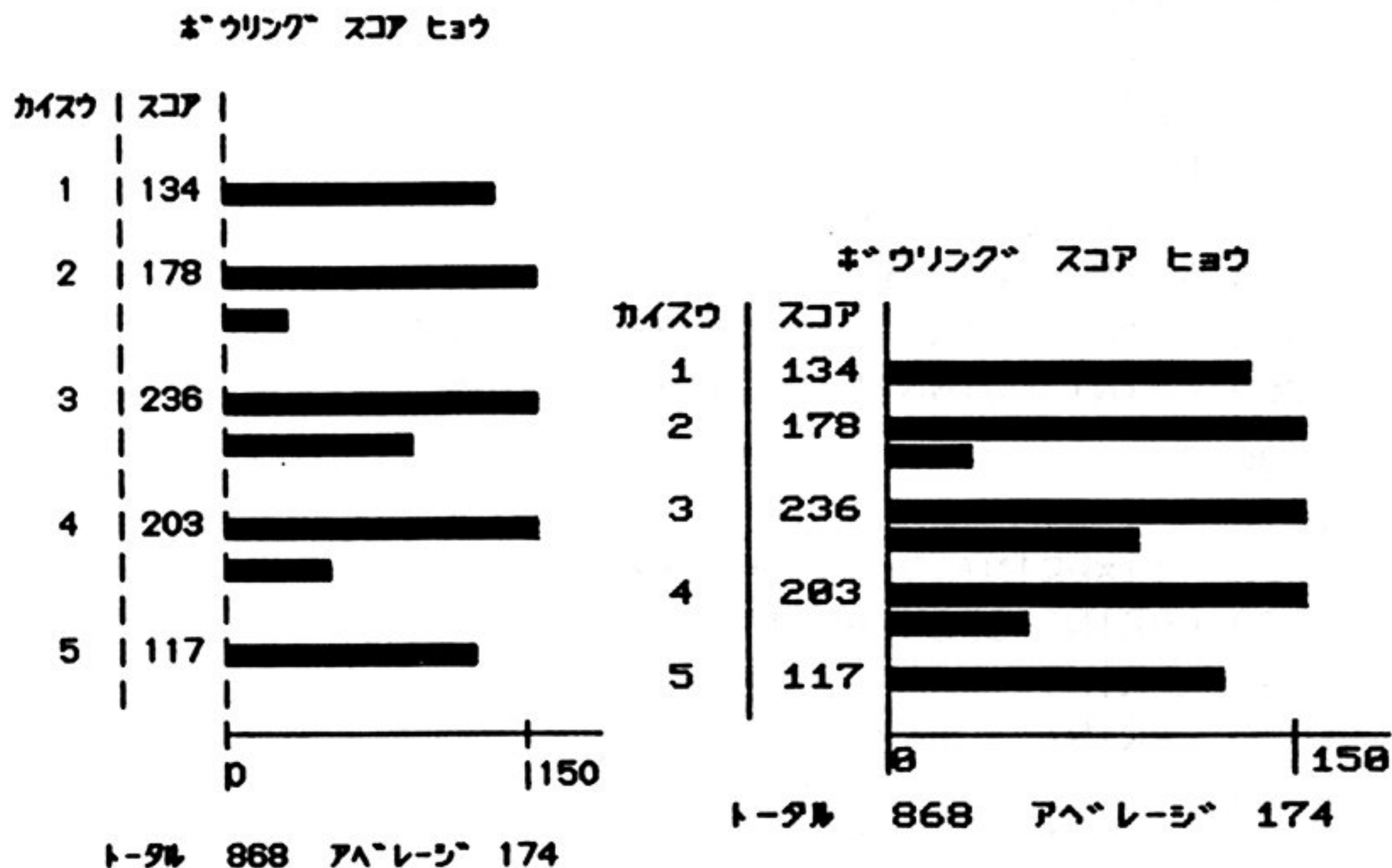
＜プログラム例＞

```
10 '*** ﾎﾞｳﾘﾝｸﾞ ｽｺｱ ﾋｮｳ ***
20 CLS:WIDTH40,20
30 LOCATE 0,20
40 INPUT"ｹﾞｰﾑｽｳ ﾉ ﾆｭｳﾘｮｸ      ",COUNT
50 DIM KAI(COUNT)
60 FOR I=1 TO COUNT
70 PRINT I;"ｶｲﾒ ﾉ ｽｺｱ      ";
80 INPUT KAI(I)
90 NEXT
100 '
110 '
120 '***CRT ﾍ ｼｭﾂﾘｮｸ***
130 WIDTH40,25
140 LOCATE 10,2:PRINT"ﾎﾞｳﾘﾝｸﾞ ｽｺｱ ﾋｮｳ"
150 PRINT
160 PRINT"  ｶｲｽｳ | ｽｺｱ |"
170 FOR I=1 TO COUNT
180 PRINT       "         |     |"
190 PRINT USING"    ##  |#### |";I,KAI(I);
200 TEN=KAI(I)/10
210 IF TEN<=15 THEN 250
220 PRINT STRING$(15,&H86)
230 TEN=TEN-15
240 PRINT "         |     |";
250 PRINT STRING$(TEN,&H86)
260 NEXT
270 PRINT       "         |     |"
280 PRINT    SPC(11);" |";
290 PRINT    STRING$(14,&H95);"+";"——"
300 PRINT    SPC(11)" |0"SPC(13)"|150"
```

```
 310 PRINT
 320 FOR I=1 TO COUNT
 330 SOGO=KAI(I)+SOGO
 340 NEXT
 350 AVE=SOGO/COUNT
 360 PRINT SPC(6);
 370 PRINT USING"ﾄｰﾀﾙ#####&&";SOGO,"   ";
 380 PRINT USING"ｱﾍﾞﾚｰｼﾞ####";AVE
1000 '
1010 '
1020 '***PRINTERﾍ ｼｭﾂﾘｮｸ***
1030 LPRINT SPC(10)"ﾎﾞｳﾘﾝｸﾞ ｽｺｱ ﾋｮｳ"
1040 LPRINT
1050 LPRINT"   ｶｲｽｳ | ｽｺｱ |"
1060 FOR I=1 TO COUNT
1070 LPRINT        "          |      |"
1080 LPRINT USING"     ##   |#### |";I,KAI(I);
1090 TEN=KAI(I)/10
1100 IF TEN<=15 THEN 1140
1110 LPRINT STRING$(15,&H86)
1120 TEN=TEN-15
1130 LPRINT "          |      |";
1140 LPRINT STRING$(TEN,&H86)
1150 NEXT
1160 LPRINT        "          |      |"
1170 LPRINT   SPC(11);" |";
1180 LPRINT   STRING$(14,&H95);"+";"——"
1190 LPRINT   SPC(11)" p"SPC(13)"|150"
1200 LPRINT:SOGO=0
1210 FOR I=1 TO COUNT
1220 SOGO=KAI(I)+SOGO
1230 NEXT
1240 AVE=SOGO/COUNT
1250 LPRINT SPC(6);
1260 LPRINT USING"ﾄｰﾀﾙ#####&&";SOGO,"   ";
1270 LPRINT USING"ｱﾍﾞﾚｰｼﾞ####";AVE
2000 HARDC 2
3000 END
```

＜出力結果＞

```
      ﾎﾞｳﾘﾝｸﾞ ｽｺｱ ﾋｮｳ

ｶｲｽｳ | ｽｺｱ |
     |      |
  1  | 134  |■■■■■■■■■■■
     |      |
  2  | 178  |■■■■■■■■■■■■■
     |      |■■■
     |      |
  3  | 236  |■■■■■■■■■■■■■
     |      |■■■■■■■■
     |      |
  4  | 203  |■■■■■■■■■■■■■
     |      |■■■■
     |      |
  5  | 117  |■■■■■■■■■■
     |      |
            +------------+---
            0            |150

    ﾄｰﾀﾙ  868  ｱﾍﾞﾚｰｼﾞ  174
```

```
          ﾎﾞｳﾘﾝｸﾞ  ｽｺｱ  ﾋｮｳ
ｶｲｽｳ | ｽｺｱ |
  1  | 134  |■■■■■■■■■■■■■■
  2  | 178  |■■■■■■■■■■■■■■■■
     |      |■■■
  3  | 236  |■■■■■■■■■■■■■■■■
     |      |■■■■■■■■■■
  4  | 203  |■■■■■■■■■■■■■■■■
     |      |■■■■■
  5  | 117  |■■■■■■■■■■■■■
            +---------------+---
            0               |150
  ﾄｰﾀﾙ   868   ｱﾍﾞﾚｰｼﾞ  174
```

14.5 OPEN 命令における指定

OPEN 命令は，プリンタに対するファイルを開きます．通常のデータ出力時は，プリンタの専用命令（LLIST，LPRINT 等）を使用すると，プログラムのステップ数の減少や扱い方が簡単になります．

しかし，ビジネスプリンタを用い，改行桁数を80桁か136の指定を行なう場合や， ESC（エスケープ）シーケンスを使用する場合の改行桁数の最大値の補正等を行なう場合，OPEN 命令によりそれらの指定が行なえます．

なお，OPEN 命令により開いたファイルは，必ず CLOSE 命令により閉じて下さい．

> OPEN "O"，ファイル番号，"LPTO：$\left(\begin{Bmatrix} S \\ W \end{Bmatrix} \begin{Bmatrix} F \\ N \end{Bmatrix}\right)$"
>
> 第1パラメータ：{S／W}
> 第2パラメータ：{F／N}

プリンタに対する OPEN 文において，オプションの第2パラメータに，F または N の記述により，プリンタの印字桁数のカウントを F-BASIC で行なうか行なわないかの指定ができます．オプションの意味を，次に示します．

(1) 第1パラメータ (SまたはW)

改行桁数を80桁または，136桁とします．

S……改行桁数を最大80桁とします．ただし，以下の場合は，80桁で改行は行ないません．

1．出力するデータが一度に80桁以上の場合

PRINT #1，"└──────────┘"

80桁以上のデータ

2．出力するデータの合計が80桁となり，さらにデータを出力した場合

PRINT #1，"└───┘"；"└───┘"；"└───┘"

40桁のデータ 40桁のデータ 30桁のデータ

上の場合ですと，1行に110桁のデータを印字します．

W……改行桁数を最大136桁とします．

(2) 第2パラメータ (FまたはN)

F……出力しようとするデータの長さが，現在のプリント位置から行の終りまでの長さより長ければ，改行して，行の先頭から出力します．

N……データが，指定した最大桁数（80桁または136桁）になるまで改行はしません．上記の場合であっても，現在のプリント位置から出力し，1行に出力されないデータは，改行後，行の先頭から出力します．

このオプション指定は，プリンタのESCシーケンスを使用したとき，プリンタヘッドの印字桁位置が変わり，予期せぬ箇所で改行が行なわれた場合に使用して下さい．

14.6 コントロールコードによるプリンタの制御

コントロールコードは，コマンドとエスケープコントロールに機能を分けることができます．

コマンドは，コードのみをプリンタへ送出することにより，そのコードに対する機能を発揮します．

エスケープコントロールは，エスケープコード (ESC：&H1B) と，それに続くデータにより，各種の機能を使用することができます．

次に，コードとそれに対する機能および使用方法を示します．

なお，使用上の詳細な規約等については，製品添付の取扱説明書を参照して下さい．

(1) コマンド

コマンド名称	対応機種 ○=可，×=不可				記号	コード(16進)	機能説明
	ドットプリンタ	シリアルドットプリンタ		ビジネスプリンタ			
	MB27404	MB27401	MB27403	MB27402			
キャリッジリターン	○	○	○	○	CR	0D	印字後ヘッドを復帰します．
ラインフィード	○	○	○	○	LF	0A	印字後改行します．
フォームフィード	○	○	○	○	FF	0C	印字後次頁の頭出しを行ないます．
バーティカルタブ	○	○	○	○	VT	0B	印字後改行します．
ホリゾンタルタブ	×	×	○	×	HT	09	1ライン上へホリゾンタルタビュレーション動作を行ないます．
ベル	○	○	○	○	BEL	07	ブザーが一定時間鳴ります．
シフトアウト	○	○	○	○	SO	0E	横2倍拡大文字を指定します．
シフトイン	○	○	○	×	SI	0F	縮小文字を指定します．
バックスペース	○	×	○	○	BS	08	プリントバッファ内の最後の文字データを消去します．
キャンセル	×	×	×	○	CAN	18	プリントバッファ内のデータをすべて消去します．
デリート	×	×	○	○	DEL	7F	プリントバッファ内の最後の文字データを消去します．
デバイスコントロール4	○	○	○	○	DC4	14	SO機能の解除を行ないます．
デバイスコントロール2	○	○	○	×	DC2	12	SI機能の解除を行ないます．
デバイスコントロール3	×	×	×	○	DC3	13	プリンタをディセレクト状態にします．
デバイスコントロール1	×	×	×	○	DC1	11	プリンタをセレクト状態にします．

(2) エスケープコントロール ESC (&H1B) POS (&H10) FS (&H1C) NUL (&H00)

コントロール名称		対応機種 ○=可, ×=不可				エスケープコントロール	機能説明
		ドットプリンタ	シリアルドットプリンタ		ビジネスプリンタ		
		MB27404	MB27401	MB27403	MB27402		
ページ長指定		○	○	○	○	ESC+C (またはテ)+n	1ページ当りの行数を指定します.
		○	○	○	×	ESC+C+$0_{(H)}$+m	1ページをインチにより指定します.
改行ピッチ指定		×	×	×	○	ESC+W (またはラ)+n	1/120インチ単位の改行を行ないます.
		×	○	○	○	ESC+A (またはチ)+n	シリアルドットプリンタ 1/72インチ単位の改行 ビジネスプリンタ 1/60インチ単位の改行
		○	○	○	○	ESC+0	ドットプリンタ 1/9インチに設定します. シリアルドットプリンタ 1/8インチに設定します. ビジネスプリンタ 16/120インチに設定します.
		○	○	○	○	ESC+2 (またはイ)	1/6インチに設定します.
改行ピッチ指定		×	×	○	×	ESC+1	7/72インチに設定します.
		×	×	○	×	ESC+3+n	1/216インチ単位の改行を行ないます.
単発改行ピッチ指定		×	×	○	○	ESC+J (またはハ)+n	シリアルドットプリンタ (MB27403) 1/216インチ単位の改行 ビジネスプリンタ 1/120インチ単位の改行
水平アドレス指定		×	×	×	○	ESC+6 (またはカ)+n1+n2	1/20インチ単位の指定を行ないます.
HTタブ (ホリゾンタルタブ)		×	×	○	×	ESC+D+n1+n2… …nk+NUL$(0)_H$	ホリゾンタルタブの設定を行ないます.
VTタブ (バーティカルタブ)	設定	×	×	×	○	ESC+Q (またはム)+n1+n2… …nX+FF	バーティカルタブ位置を設定します.
	解除	×	×	×	○	ESC+Q (またはム)+FF	バーティカルタブ位置を解除します.
印字桁数の設定		×	×	○	×	ESC+Q+n	文字データの印字桁数を設定します.
スキップパーフォレーション	設定	○	○	○	○	ESC+N (またはホ)+n	スキップする行数を設定します.
	解除	○	○	○	○	ESC+O (またはマ)	スキップパーフォレーションを解除します.
片方向印字	設定	×	×	×	○	ESC+G (またはヌ)	片方向印字の設定を行ないます.
	解除	×	×	×	○	ESC+T (またはヤ)	片方向印字の解除を行ないます.
二重印字	設定	×	×	○	×	ESC+G	文字データの二重印字を指定します.
	解除	×	×	○	×	ESC+H	二重印字の解除を行ないます.

コントロール名称	対応機種 ○=可, ×=不可				エスケープコントロール	機能説明
	ドットプリンタ	シリアルドットプリンタ		ビジネスプリンタ		
	MB27404	MB27401	MB27403	MB27402		
8ドット標準ビットイメージ	○	○	○	○	ESC+K(またはヒ)+n1+n2 +データ	8ドットによるビットイメージを出力します.
8ドット倍密度ビットイメージ	○	○	○	○	ESC+L(またはフ)+n1+n2 +データ	8ドットによる横方向倍密度ビットイメージを出力します.
16ドットビットイメージ	×	×	×	○	ESC+4(またはエ)+n1+n2 +データ	16ドットによるビットイメージを出力します.
グラフィックデータ 回数指定	○	×	×	×	FS+n+データ	グラフィックデータの繰り返し回数を指定します.
レギュラモード	×	×	×	○	ESC+R(またはメ)	高印字品質を指定します.
ドラフトモード	×	×	×	○	ESC+D(またはト)	高速印字を指定します.
パイカピッチ	×	×	×	○	ESC+P(またはミ)	10文字/インチのピッチを指定します.
エリートピッチ	×	×	×	○	ESC+E(またはナ)	12文字/インチのピッチを指定します.
スーパースクリプト	×	×	○	○	シリアルドットプリンタ (MB27403) ESC+S+0 ビジネスプリンタ ESC+U(またはユ)	スーパースクリプトを指定します.
サブスクリプト	×	×	○	○	シリアルドットプリンタ (MB27403) ESC+S+1 ビジネスプリンタ ESC+V(またはヨ)	サブスクリプトを指定します.
スーパー/サブスクリプト解除	×	×	○	○	シリアルドットプリンタ (MB27403) ESC+T ビジネスプリンタ ESC+F(またはニ)	スーパースクリプトおよびサブスクリプトを解除します.
強調文字指定 設定	×	×	○	×	ESC+E	強調文字を指定します.
強調文字指定 解除	×	×	○	×	ESC+F	強調文字を解除します.
アンダーライン 設定	×	×	○	○	シリアルドットプリンタ (MB27403) ESC+−+1 ビジネスプリンタ ESC+I(またはノ)	指定から解除までのデータへ, アンダーラインを付けます.
アンダーライン 解除	×	×	○	○	シリアルドットプリンタ (MB27403) ESC+−+0 ビジネスプリンタ ESC+X(またはリ)	
インチスペース	×	×	×	○	ESC+S(またはモ)+n	文字データの後へ1/60インチ単位により, スペースを設定.
倍幅拡大文字	○	○	×	○	ESC+SO	文字データを, 横方向へ2倍拡大して出力を行ないます(SO機能).
倍幅拡大文字 設定	×	×	○	×	ESC+W+1	
倍幅拡大文字 解除	×	×	○	×	ESC+W+0	横2倍拡大文字設定(SO機能)を解除します.

コントロール名称	対応機種　○＝可，×＝不可				エスケープコントロール	機能説明
	ドットプリンタ	シリアルドットプリンタ		ビジネスプリンタ		
	MB27404	MB27401	MB27403	MB27402		
縮小文字を指定します.	○	○	×	×	ESC+SI	縮小文字を指定します.(SI機能)
	○	×	×	×	ESC+M	
プリント開始位置指定	○	×	×	×	POS+n1+n2	キャラクタ単位により指定します.
	○	×	×	×	ESC+POS+n1+n2	ドット単位により指定します.
単票吸入	×	×	×	○	DC2+O(またはマ)	インサータ使用時，単票を吸入します.
単票排出	×	×	×	○	DC2+P(またはミ)	インサータ使用時，単票を排出します.
紙検出信号無効化	○	○	○	×	ESC+8	紙検出器からの信号を無視します.
紙検出信号有効化	○	○	○	×	ESC+9	紙検出器からの信号を有効とします.

14.6.1 コマンド

コントロールコードにおいて，コマンドの使用例および機能についての説明を行ないます.

(1) CR（キャリッジリターン）

プリンタバッファ内のデータを始めから印字した後，プリンタヘッドの復帰を行ないます.

<プログラム例>

```
10 LPRINT"AAAA";CHR$(&H0D);
20 LPRINT"BBBB";CHR$(&H0D);
30 LPRINT"CCCC";CHR$(&H0D);
```

上の例において，データ印字後，復帰を行ないますので，“A”の上へ“B”と“C”を重ねて印字を行ないます.

(2) LF（ラインフィード）

プリンタバッファ内のデータを印字した後，プリンタヘッドの復帰と改行を行ないます.

<プログラム例>

```
10 LPRINT"AAAA";CHR$(&H0A);
20 LPRINT"BBBB";CHR$(&H0A);
30 LPRINT"CCCC";CHR$(&H0A);
```

<印字結果>

```
AAAA
BBBB
CCCC
```

(3) FF(フォームフィード)

プリンタバッファ内のデータを印字後，次のページの先頭位置まで用紙を送ります．

```
LPRINT CHR$ (&H0C)
```

(4) VT(バーティカルタブ)

ラインフィードと同様，印字後，復帰と改行を行ないます．

<プログラム例>

```
10 LPRINT"AAAA";CHR$(&H0B);
20 LPRINT"BBBB";CHR$(&H0B);
30 LPRINT"CCCC";CHR$(&H0B);
```

<印字結果>

```
AAAA
BBBB
CCCC
```

(5) HT(ホリゾンタルタブ)

タブ設定(初期値は8桁ごと)にタブ移動を行ないます．

<プログラム例>

```
10 REM ***ﾎﾘｿﾞﾝﾀﾙﾀﾌﾞ***
20 LPRINT "A";CHR$(&H09);
30 LPRINT "B";CHR$(&H09);
40 LPRINT "C"
```

<印字結果>

```
A       B       C
└───────┘└──────┘
  8桁      8桁
```

(6) BEL(ベル)

プリンタに持つブザーを一定時間鳴らせます．

```
LPRINT CHR$ (&H07)
```

(7) SO(シフトアウト)

文字データを横方向へ2倍拡大し出力を行ないます．この指定の解除は，DC4(デバイスコントロール4)または，改行動作により行ないます．

<プログラム例>

```
10 LPRINT"AAAA";
20 LPRINT CHR$(&HOE); ←——SOコードを入力します.
30 LPRINT"BBBB";
40 LPRINT CHR$(&H14); ←——SO機能の解除をDC4により行ないます.
50 LPRINT"CCCC"
```

<印字結果>

AAAABBBBCCCC

(8) SI(シフトイン)

縮小文字の指定を行ないます．この指定の解除は，DC2(デバイスコントロール2)により解除を行ないます．

縮小文字状態において，SO(シフトアウトコード)を入力することにより縮小文字の横方向2倍拡大文字を出力します．

<プログラム例>

```
10 LPRINT"AAAA";
20 LPRINT CHR$(&HOF); ←—— SIコードを入力します.
30 LPRINT"BBBB";
40 LPRINT CHR$(&HOE); ←—— SOコードを入力します.
50 LPRINT"CCCC"
60 LPRINT CHR$(&H14); ←—— SO機能の解除を行ないます.
                              (DC4による)
70 LPRINT"DDDD";
80 LPRINT CHR$(&H12); ←—— SI機能の解除を行ないます.
                              (DC2による)
90 LPRINT"EEEE"
```

<印字結果>

AAAABBBBCCCCDDDDEEEE

(9) BS(バックスペース)

プリントバッファにおいて，BS コードの直前にある文字データを消去します．

<プログラム例>

```
10 LPRINT"AAAA";
20 LPRINT CHR$(&H08);
30 LPRINT"BBBB"
```

<印字結果>

```
AAABBBB
```

(10) CAN(キャンセル)

プリントバッファ内のデータを消去します．ただし，エスケープコントロール等の指定は消去されません．

<プログラム例>

```
10 LPRINT"AAAA";
20 LPRINT CHR$(&H18);
30 LPRINT"BBBB"
```

<印字結果>

```
BBBB
```

(11) DC4 (デバイスコントロール4)，DC2 (デバイスコントロール2)

DC4は，SOコードの解除を行ない，DC2は，SIコードの解除を行ないます．プログラム例は，(7)および(8)を参照して下さい．

(12) DC3(デバイスコントロール3)，DC1(デバイスコントロール1)

これらのコマンドは，プリンタをON-LINE (DC1) にするか，またはOFF-LINE (DC3) にします．

また，扱いが異なり，以下のようにPOKE(ポーク)命令により，直接メモリへコマンドのコードを書きます．

```
10 REM ***セレクト***
20 POKE &HFD01,&H11 ←── コマンドのコードを送出します.(DC1)
30 POKE &HFD00,&H00 ←┐
40 POKE &HFD00,&H40 ←┘ コマンドを有効とする処理です.
```

ディセレクト状態にする場合は，以下のように入力します．

POKE &HFD01, &H13

14.6.2 エスケープコントロール

(1) ページ長指定　ESC+C(またはテ)+n

1ページにおける行数の指定を行ないます．この指定を行なうと，そこがページの頭となります．また，この指定によるページの長さは，改行ピッチを変更することにより変化します．リセット時は66行に設定されます．

<プログラム例>

```
10 REM ***ﾍﾟｰｼﾞﾁｮｳ ｼﾃｲ ｷﾞｮｳｽｳ***
20 LPRINT CHR$(&H1B);"C";
30 LPRINT CHR$(&H30)
```

次に，行数ではなく，インチによりページの長さを決める方法を述べます．この方法であれば，改行ピッチの変更に伴なうページの変化は起こりません．

ESC+C+0(16進)+m　　m：インチ数

<プログラム例>

```
10 REM ***ﾍﾟｰｼﾞﾁｮｳ ｼﾃｲ ｲﾝﾁ***
20 LPRINT CHR$(&H1B);"C";CHR$(&H00);
30 LPRINT CHR$(10)
```

(2) 改行ピッチの指定

改行する際の，行間隔を指定するために，改行ピッチ指定を行ないます．リセット時は1/6インチに設定されています．

① ビジネスプリンタ1/120インチ単位　　ESC+W(またはラ)+n

<プログラム例>

```
10 REM ***ｶｲｷﾞｮｳ 1/120***
20 LPRINT CHR$(&H1B);"W";CHR$(&H01)
30 LPRINT "FUJITSU MICRO 7"
40 LPRINT "FUJITSU MICRO 7"
50 LPRINT "FUJITSU MICRO 7"
60 LPRINT "FUJITSU MICRO 7"
70 LPRINT "FUJITSU MICRO 7"
80 LPRINT "FUJITSU MICRO 7"
```

<印字結果>

FUJITSU MICRO 7

② シリアルドットプリンタ 1/72

ビジネスプリンタ 1/60インチ単位　　ESC＋A（またはチ）＋n

＜プログラム例＞

```
10 REM ***ｶｲｷﾞｮｳ 1/60***
20 LPRINT CHR$(&H1B);"A";CHR$(&H01)
30 LPRINT "FUJITSU MICRO 7"
40 LPRINT "FUJITSU MICRO 7"
50 LPRINT "FUJITSU MICRO 7"
60 LPRINT "FUJITSU MICRO 7"
70 LPRINT "FUJITSU MICRO 7"
80 LPRINT "FUJITSU MICRO 7"
```

＜印字結果＞

```
FUJITSU MICRO 7
```

③ ドットプリンタ 1/9インチ，シリアルドットプリンタ 1/8インチ，ビジネスプリンタ 16/120インチに指定　　ESC＋0

＜プログラム例＞

```
10 REM ***ｶｲｷﾞｮｳ 16/120***
20 LPRINT CHR$(&H1B);"0";
30 LPRINT "FUJITSU MICRO 7"
40 LPRINT "FUJITSU MICRO 7"
50 LPRINT "FUJITSU MICRO 7"
60 LPRINT "FUJITSU MICRO 7"
70 LPRINT "FUJITSU MICRO 7"
80 LPRINT "FUJITSU MICRO 7"
```

＜印字結果＞

```
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
```

④ 1/6インチ指定　　ESC+2(またはイ)

全機種の初期状態の指定です.

<プログラム例>

```
10 REM ***ｶｲｷﾞｮｳ 1/6***
20 LPRINT CHR$(&H1B);"2";
30 LPRINT "FUJITSU MICRO 7"
40 LPRINT "FUJITSU MICRO 7"
50 LPRINT "FUJITSU MICRO 7"
60 LPRINT "FUJITSU MICRO 7"
70 LPRINT "FUJITSU MICRO 7"
80 LPRINT "FUJITSU MICRO 7"
```

<印字結果>

```
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
FUJITSU MICRO 7
```

(3) 単発改行ピッチ　　ESC+J(またはハ)+n

シリアルドットプリンタ(MB27403)(ビジネスプリンタ)において, 1回の改行指定が行なえます.

<プログラム例>

```
10 REM ***ﾀﾝﾊﾟﾂ ｶｲｷﾞｮｳ ﾋﾟｯﾁ***
20 LPRINT CHR$(&H1B);"A";CHR$(&H01)
30 LPRINT "FUJITSU MICRO 7"
40 LPRINT "FUJITSU MICRO 7"
50 LPRINT "FUJITSU MICRO 7"
60 LPRINT CHR$(&H1B);"J";CHR$(&H0F)
70 LPRINT "FUJITSU MICRO 7"
80 LPRINT "FUJITSU MICRO 7"
90 LPRINT "FUJITSU MICRO 7"
```

<印字結果>

```
FUJITSU MICRO 7
FUJITSU MICRO 7
```

(4) 水平アドレス指定　ESC+6（またはカ）+n1+n2

ビジネスプリンタにおいて，1行の印字開始位置を1/20インチ単位により設定できます．

<プログラム例>

```
10 REM ***ｽｲﾍｲ ｱﾄﾞﾚｽ***
20 LPRINT "FUJITSU MICRO 7"
30 LPRINT CHR$(&H1B);"6";CHR$(&H00);CHR$(&H14);
40 LPRINT "FUJITSU MICRO 7"
50 LPRINT "FUJITSU MICRO 7"
```

";"（セミコロン）が必要です．

2バイトバイナリデータにより，16進の0014は20（10進）となりますので，20／20，つまり，1インチの指定となります．

<印字結果>

```
FUJITSU MICRO 7
            FUJITSU MICRO 7
FUJITSU MICRO 7
```

(5) HTタブ(ホリゾンタルタブ)　ESC+D+n1+n2……nK+NUL

シリアルドットプリンタ(MB27403)において，水平方向のタブ設定を行ないます．

<プログラム例>

```
10 REM ***ﾎﾘｿﾞﾝﾀﾙﾀﾌﾞ ｾｯﾃｲ***
20 LPRINT CHR$(&H1B);"D";CHR$(&H05);CHR$(&H0A);CHR$(&H00)
30 LPRINT "A";CHR$(&H09);
40 LPRINT "B";CHR$(&H09);
50 LPRINT "C"
```

<印字結果>

```
A    B    C
|____|____|
 5桁  5桁
```

(6) VTタブ(バーティカルタブ)　ESC+Q+n1+n2……nX+FF

ビジネスプリンタにおいて，行数におけるタブ設定を行ないます．

<プログラム例>

```
10 REM ***ｽｲﾁｮｸ ｱﾄﾞﾚｽ***
20 LPRINT "FUJITSU MICRO 7"
30 LPRINT CHR$(&H1B);"Q";
40 LPRINT CHR$(&H05);CHR$(&H0A);   ←5行目と10行目にタブを設定します．
50 LPRINT CHR$(&HFF);              ←タブ設定の終了を示します．
60 LPRINT "FUJITSU MICRO 7";CHR$(&H0B);
70 LPRINT "FUJITSU MICRO 7";CHR$(&H0B);
80 LPRINT "FUJITSU MICRO 7";CHR$(&H0B);
90 LPRINT "FUJITSU MICRO 7";CHR$(&H0B);
100 LPRINT CHR$(&H1B);"Q";CHR$(&HFF)  ←タブ指定を解除します．
```

<印字結果>

```
FUJITSU MICRO 7 ←──行番号 20 による出力です.
FUJITSU MICRO 7 ←──行番号 60 による出力であり, この印字後,
                     次のタブ位置へ移動します.

FUJITSU MICRO 7 ←──行番号 70 による出力です



FUJITSU MICRO 7 ←──行番号 80 と 90 による出力です. 行番号 80 による印字
                     後, タブ位置は 11 行目以降設定されていないので, 行
                     番号 90 による出力を重ねて印字します.
```

(7) 印字桁数の指定　　ESC+Q+n

シリアルドットプリンタ (MB27403) において, 1行に印字する桁数を指定できます.

<プログラム例>

```
10 REM ***ｲﾝｼﾞ ｹﾀｽｳ ｼﾃｲ***
20 LPRINT CHR$(&H1B);"Q";CHR$(&H03);←3桁に指定します.
30 LPRINT"AAAAA"
40 LPRINT CHR$(&H1B);"Q";CHR$(&H06);←6桁に指定します.
50 LPRINT"BBBBBBBBBB"
60 LPRINT CHR$(&H1B);"Q";CHR$(&H50)  ←初期設定の 80 桁に戻します.
```

<印字結果>

```
AAA
AA
BBBBBB
BBBB
```

(8) スキップパーフォレーション

連続用紙には, 1ページごとにミシン目があり, 切り離すことができます.

ところが, 印字を行ない, このミシン目にまたがると印字が切れてしまいます. そのため, スキップパーフォレーション機能を用い, 1ページの最後の数行を送ってしまうことができます.

① スキップパーフォレーションの設定　　ESC+N(またはホ)+n

```
LPRINT CHR$(&H1B)+"N"+CHR$(&H05)
```

送る行数を指定します. (CHR$(&H05))
この場合は, 5行とします.

② スキップパーフォレーション解除　　ESC+O(またはマ)

```
LPRINT CHR$(&H1B)+"O"
```

(9) 片方向印字指定

グラフィックパターンをプリンタへ印字する際，プリンタヘッドは，左から右への移動方向のみにより印字を行ないます．

これに対し，キャラクタパターンを印字する場合は，双方向（左—右）により印字を行ないます．

この，キャラクタパターン印字の際の片方向印字指定を以下に示します．

① 指定を行なう場合　ESC+G（またはヌ）

```
LPRINT CHR$(&H1B);"G"
```

② 解除する場合　ESC+T（またはヤ）

```
LPRINT CHR$(&H1B);"T"
```

(10) 二重印字指定

シリアルドットプリンタ（MB27403）において，印字データを二重に印字して，濃い印字が得られます．

① 指定を行なう場合　ESC+G

```
LPRINT CHR$(&H1B);"G"
```

② 解除する場合　ESC+H

```
LPRINT CHR$(&H1B);"H"
```

(11) 8ドット標準ビットイメージ　ESC+K（またはヒ）+n1+n2+データ

8ドットによるビットイメージの出力を行ないます．

ヘッドピン

● 8
○ 7
○ 6
○ 5　ピン番号
○ 4
○ 3
○ 2
● 1

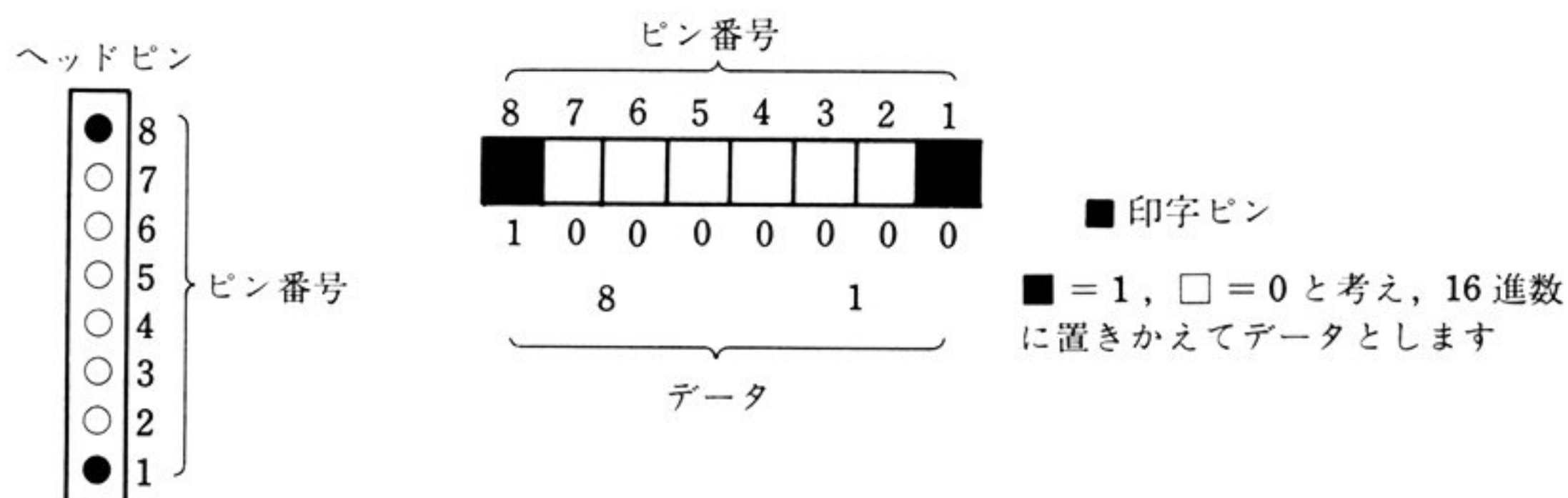
ピン番号

8	7	6	5	4	3	2	1
■	□	□	□	□	□	□	■
1	0	0	0	0	0	0	0

8　1

データ

■ 印字ピン

■ = 1，□ = 0 と考え，16進数に置きかえてデータとします

前図に示すように，横方向1ドットの8ドットビットイメージは次のようになります．

LPRINT CHR$(&H1B);"K";CHR$(&H01);CHR$(&H00);CHR$(&H81)

n1　n2　データ

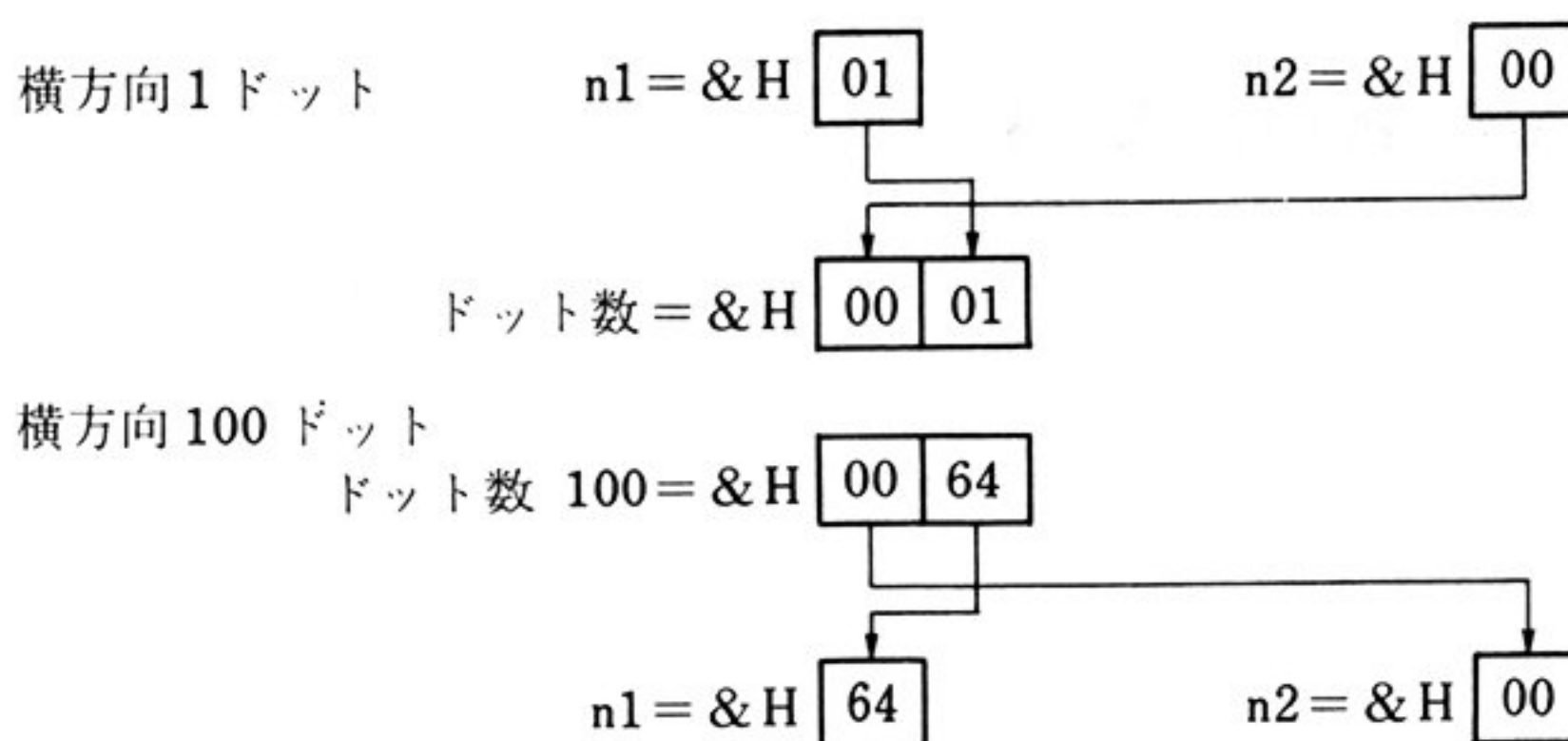
横方向のドット数
(印字回数)

横方向のドット数は以下のように指定します．

横方向1ドット　n1＝&H 01　n2＝&H 00

ドット数＝&H 00 01

横方向100ドット
ドット数 100＝&H 00 64

n1＝&H 64　n2＝&H 00

<プログラム例>

```
10 REM ***8ビット ヒョウジュン ビットイメージ***
20 LPRINT CHR$(&H1B);"K";CHR$(&H50);CHR$(0);
30     FOR J=1 TO &H50
40         LPRINT CHR$(&H81);
50     NEXT
60 LPRINT
```

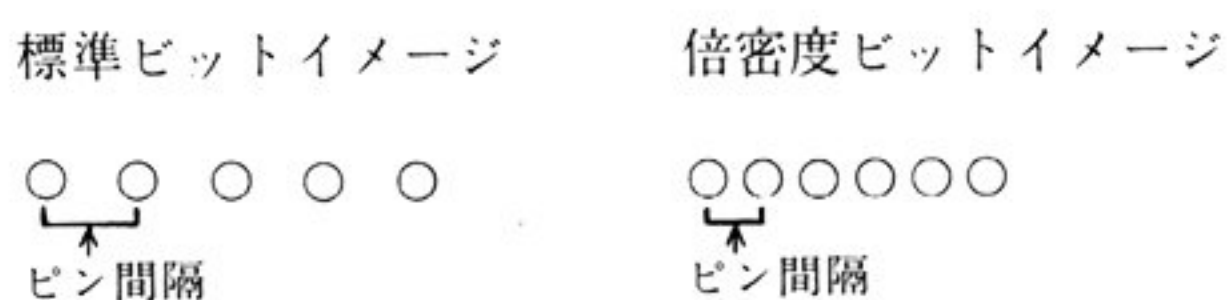
<印字結果>

(12) 8ドット倍密度ビットイメージ　ESC＋L(またはフ)＋n1＋n2＋データ

8ドットによる横方向に対する倍密度のビットイメージ出力を行ないます．8ドット標準ビットイメージのドット間隔を，横方向へ1ピン分つまったビットイメージです．

標準ビットイメージ　倍密度ビットイメージ

○ ○ ○ ○ ○　○○○○○○

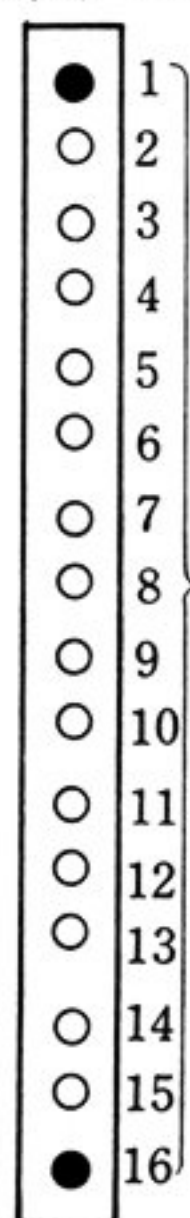
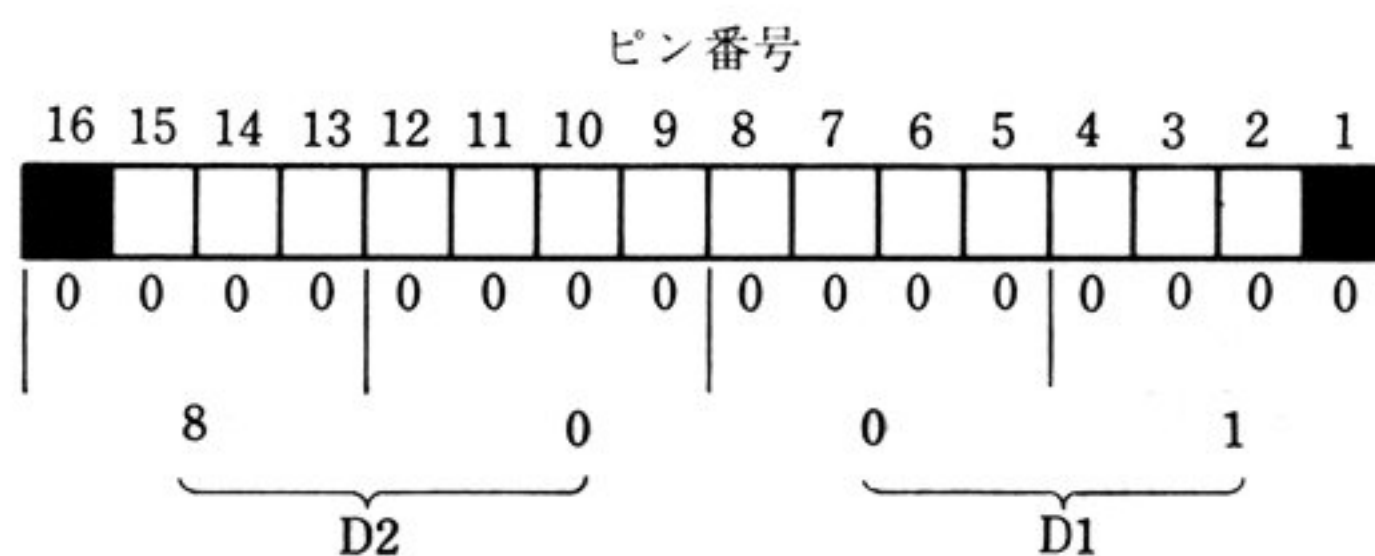
ピン間隔　ピン間隔

印字するイメージおよび印字回数は，8ドット標準ビットイメージと同様です．

<プログラム例>

```
10 REM ***8ﾋﾞｯﾄ ﾊﾞｲﾐﾂ ﾋﾞｯﾄｲﾒｰｼﾞ***
20 LPRINTCHR$(&H1B);"L";CHR$(0);CHR$(1);
30    FOR J=0 TO &HFF
40        LPRINT CHR$(&HFF);
50    NEXT J
60 LPRINT
```

<印字結果>

(13) 16ドットビットイメージ　　ESC+4(またはエ)+n1+n2+データ

ビジネスプリンタは，印字するための印字ピンを，16本持っており，このピンにより，16ドットによるビットイメージを出力できます．

□＝0，■＝1と考え，16進数に置きかえ，データとします

上図の場合，横方向1ドットの16ドットビットイメージは次のようになります．

LPRINT CHR$(&H1B);"4";CHR$(&H00);CHR$(&H01);

n1　n2

横方向のドット数（印字回数）

CHR$(&H01);CHR$(&H80)

上図に示すデータ

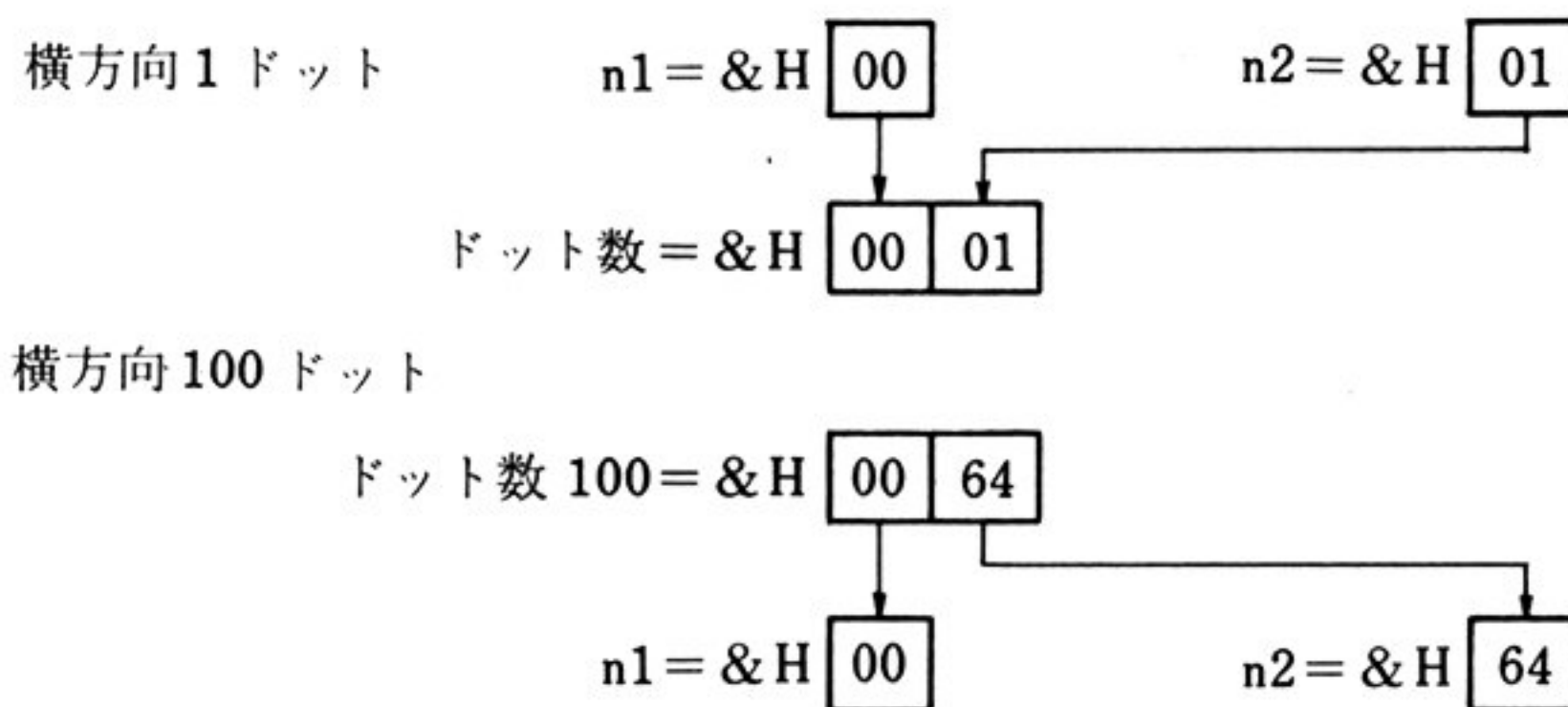
横方向のドット数は以下のように指定します.

横方向1ドット　n1＝&H 00　n2＝&H 01

ドット数＝&H 00 01

横方向100ドット

ドット数100＝&H 00 64

n1＝&H 00　n2＝&H 64

以下に16ドットビットイメージの出力例を示します.

<プログラム例>

```
10 REM ***16ト゛ット ヒ゛ットイメーシ゛***
20 LPRINTCHR$(&H1B);"4";CHR$(1);CHR$(0);
30    FOR J=0 TO &HFF
40        LPRINT CHR$(&H01);CHR$(&H80);
50    NEXT J
60 LPRINT
```

<印字結果>

(14)　グラフィックデータ回数指定　　FS＋n＋データ

ドットプリンタにおいて，8ドットのグラフィックデータを繰返し，印字する指定ができます.
データの指定は8ドットビットイメージと同様です.

<プログラム例>

```
10 REM ***ク゛ラフィック テ゛ータ***
20 LPRINT CHR$(&H1C);CHR$(&H50);CHR$(&H81)
```

<印字結果>

(15) ビジネスプリンタに関する出力例

以下に,ビジネスプリンタに関する出力例を示します．出力するコントロール名称は次の通りです．

- レギュラモード　ESC+R ⎫ 初期設定はドラフトモードです．
- ドラフトモード　ESC+D ⎭
- パイカピッチ　ESC+P ⎫ 初期設定はパイカピッチです．
- エリートピッチ　ESC+E ⎭
- スーパースクリプト　ESC+U
- サブスクリプト　ESC+V

① レギュラモード（高品質印字），パイカピッチ（ANK普通文字を10文字/インチで印字），普通の文字の大きさで出力した例．

<プログラム例>

```
10 REM ***ﾚｷﾞｭﾗﾓｰﾄﾞ・ﾊﾟｲｶﾋﾟｯﾁ***
20 LPRINT CHR$(&H1B);"R"
30 LPRINT"ﾚｷﾞｭﾗﾓｰﾄﾞ・ﾊﾟｲｶﾋﾟｯﾁ"
40        FOR I=&H30 TO &H5A
50            LPRINT CHR$(I);
60        NEXT
```

<印字結果>

```
ﾚｷﾞｭﾗﾓｰﾄﾞ・ﾊﾟｲｶﾋﾟｯﾁ
0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

② ドラフトモード（高速度印字），パイカピッチ（ANK普通文字を10文字/インチで印字），普通の文字の大きさで出力した例．

<プログラム例>

```
10 REM ***ﾄﾞﾗﾌﾄﾓｰﾄﾞ・ﾊﾟｲｶﾋﾟｯﾁ***
20 LPRINT CHR$(&H1B);"D"
30 LPRINT"ﾄﾞﾗﾌﾄﾓｰﾄﾞ・ﾊﾟｲｶﾋﾟｯﾁ"
40        FOR I=&H30 TO &H5A
50            LPRINT CHR$(I);
60        NEXT
```

<印字結果>

```
ﾄﾞﾗﾌﾄﾓｰﾄﾞ・ﾊﾟｲｶﾋﾟｯﾁ
0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

③ レギュラモード（高品質印字），パイカピッチ（ANK普通文字で10文字/インチで印字），倍幅拡大文字（横方向に2倍拡大した文字）の出力例

<プログラム例>

```
10 REM ***ﾚｷﾞｭﾗﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ･ﾊﾞｲﾊﾊﾞ***
20 LPRINT CHR$(&H1B);"R"
30 LPRINT CHR$(&H0E);
40 LPRINT"ﾚｷﾞｭﾗﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ･ﾊﾞｲﾊﾊﾞ"
50 LPRINT CHR$(&H0E);
60       FOR I=&H30 TO &H45
70           LPRINT CHR$(I);
80       NEXT
```

<印字結果>

```
ﾚｷﾞｭﾗﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ･ﾊﾞｲﾊﾊﾞ
0123456789:;<=>?@ABCDE
```

④ ドラフトモード（高速度印字），パイカピッチ（ANK普通文字を10文字/インチで印字），倍幅拡大文字（横方向に2倍拡大した文字）の出力例

<プログラム例>

```
10 REM ***ﾄﾞﾗﾌﾄﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ･ﾊﾞｲﾊﾊﾞ***
20 LPRINT CHR$(&H1B);"D"
30 LPRINT CHR$(&H0E);
40 LPRINT"ﾄﾞﾗﾌﾄﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ･ﾊﾞｲﾊﾊﾞ"
50 LPRINT CHR$(&H0E);
60       FOR I=&H30 TO &H45
70           LPRINT CHR$(I);
80       NEXT
```

<印字結果>

```
ﾄﾞﾗﾌﾄﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ･ﾊﾞｲﾊﾊﾞ
0123456789:;<=>?@ABCDE
```

⑤ レギュラモード（高品質印字），パイカピッチ（ANK普通文字で10文字/インチで印字）で普通文字と倍幅拡大文字（横方向に2倍拡大した文字）を混在させた出力例

<プログラム例>

```
10 REM ***ﾚｷﾞｭﾗﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ
20 REM             ﾉｰﾏﾙ/ﾊﾞｲﾊﾊﾞ･ｺﾝｻﾞｲ***
30 LPRINT CHR$(&H1B);"R";
40 LPRINT CHR$(&H0E);"F";
50 LPRINT CHR$(&H14);"ujitsu ";
60 LPRINT CHR$(&H0E);"M";
70 LPRINT CHR$(&H14);"icro ";
80 LPRINT CHR$(&H0E);"7"
```

<印字結果>

Ｆujitsu Ｍicro ７

⑥ ドラフトモード（高速度印字），パイカピッチ（ANK 普通文字で10文字/インチで印字），普通文字と倍幅拡大文字（横方向に2倍拡大した文字）を混在させた出力例

<プログラム例>

```
10 REM ***ﾄﾞﾗﾌﾄﾓｰﾄﾞ･ﾊﾟｲｶﾋﾟｯﾁ
20 REM             ﾉｰﾏﾙ/ﾊﾞｲﾊﾊﾞ･ｺﾝｻﾞｲ***
30 LPRINT CHR$(&H1B);"D";
40 LPRINT CHR$(&H0E);"F";
50 LPRINT CHR$(&H14);"ujitsu ";
60 LPRINT CHR$(&H0E);"M";
70 LPRINT CHR$(&H14);"icro ";
80 LPRINT CHR$(&H0E);"7"
```

<印字結果>

Ｆujitsu Ｍicro ７

⑦ レギュラモード（高品質印字），サブスクリプト文字（縮小文字を16ピンドットマトリックスの下側8ドットで印字させる）の出力例及び応用例

<プログラム例>

```
10 REM ***ﾚｷﾞｭﾗﾓｰﾄﾞ･ｻﾌﾞｽｸﾘﾌﾟﾄ***
20 LPRINT CHR$(&H1B);"R";
30 LPRINT CHR$(&H1B);"V";
40       FOR I=&H30 TO &H5A
50           LPRINT CHR$(I);
60       NEXT
70 LPRINT CHR$(&H1B);"F"
```

<印字結果>

0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ

＜サブスクリプトの応用例＞

```
10 REM ***    ｵｳﾖｳﾚｲ    ***
20 LPRINT CHR$(&H1B);"R";
30 LPRINT"ﾐｽﾞ ﾉ ｾｲﾌﾞﾝ ﾊ H";
40 LPRINT CHR$(&H1B);"V";
50 LPRINT"2";
60 LPRINT CHR$(&H1B);"F";
70 LPRINT"O ﾃﾞｽ "
```

＜印字結果＞

ﾐｽﾞ ﾉ ｾｲﾌﾞﾝ ﾊ H_2O ﾃﾞｽ

⑧ レギュラモード（高品質印字），スーパースクリプト文字（縮小文字16ピンドットマトリックスの上側8ドットで印字させる）の出力例及び応用例

＜プログラム例＞

```
10 REM ***ﾚｷﾞｭﾗﾓｰﾄﾞ･ｽｰﾊﾟｰｽｸﾘﾌﾟﾄ***
20 LPRINT CHR$(&H1B);"R";
30 LPRINT CHR$(&H1B);"U";
40       FOR I=&H30 TO &H5A
50            LPRINT CHR$(I);
60       NEXT
70 LPRINT CHR$(&H1B);"F"
```

＜印字結果＞

0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ

＜応用例＞

```
10 REM ***    ｵｳﾖｳﾚｲ(1)***
20 A=.00001:B=.00002
30 LPRINT A;"*";B;"=";:C=A*B
40 C$=STR$(C)
50 D=INSTR(C$,"E")
60 LPRINT LEFT$(C$,D-1);CHR$(&H1B);"U";
70 LPRINT MID$(C$,D+1,4);CHR$(&H1B);"F"
```

＜印字結果＞

.00001 * .00002 = 2^{-10}

<応用例2>

```
10 REM ***    オウヨウレイ(2)***
20 A=10000:B=20000
30 LPRINT A;"*";B;"=";:C=A*B
40 C$=STR$(C)
50 D=INSTR(C$,"E")
60 LPRINT LEFT$(C$,D-1);CHR$(&H1B);"U";
70 LPRINT MID$(C$,D+1,4);CHR$(&H1B);"F"
```

<印字結果>

10000 * 20000 = 2^{+08}

(16) シリアルドットプリンタ(MB27403)に関する出力例

以下にシリアルドットプリンタ(MB27403)に関する出力例を示します．出力するコントロール名称を以下に示します．

- スーパースクリプト　ESC+S+0
- サブスクリプト　ESC+S+1
- 強調文字指定　ESC+E

<プログラム例>

```
10 REM ***スーパー/サブスクリプト***
20 LPRINT CHR$(&H1B);"S";CHR$(&H0);
30 LPRINT"ABC"
40 LPRINT CHR$(&H1B);"T";
50 LPRINT"DEF"
60 LPRINT CHR$(&H1B);"S";CHR$(&H1);
70 LPRINT"GHI"
80 LPRINT CHR$(&H1B);"T";
90 LPRINT CHR$(&H1B);"H";
100 LPRINT"JKL"
```

<印字結果>

^{ABC}DEF$_{GHI}$JKL

<プログラム例>

```
10 REM ***ニジュウ インジ***
20 LPRINT CHR$(&H1B);"G";
30 LPRINT"ABCDEFGH";
40 LPRINT CHR$(&H1B);"H"
50 REM ***キョウチョウ インジ***
60 LPRINT CHR$(&H1B);"E";
70 LPRINT"ABCDEFGH";
80 LPRINT CHR$(&H1B);"F"
```

<印字結果>

```
ABCDEFGH
ABCDEFGH
```

(17) アンダーライン

シリアルドットプリンタ（MB27403）および，ビジネスプリンタにおいて，文字データへアンダーラインを付けることができます.

① シリアルドットプリンタの場合

<プログラム例>

```
10 REM ***ｱﾝﾀﾞｰﾗｲﾝ***
20 LPRINT CHR$(&H1B);"G";
30 LPRINT CHR$(&H1B);"-";CHR$(&H1);
40        FOR I=&H30 TO &H5A
50            LPRINT CHR$(I);
60        NEXT
70 LPRINT CHR$(&H1B);"-";CHR$(&H0);
80 LPRINT CHR$(&H1B);"H"
```

<印字結果>

<u>0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ</u>

② ビジネスプリンタの場合

<プログラム例>

```
10 REM ***ｱﾝﾀﾞｰﾗｲﾝ***
20 LPRINT CHR$(&H1B);"R";
30 LPRINT CHR$(&H1B);"I";
40        FOR I=&H30 TO &H5A
50            LPRINT CHR$(I);
60        NEXT
70 LPRINT CHR$(&H1B);"X"
```

<印字結果>

<u>0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ</u>

(18) インチスペース ESC+S（またはモ）+n

ビジネスプリンタにおいて，文字データの後へ1/60単位によりスペースを設定できます.

<プログラム例>

```
10 REM ***ｲﾝﾁｽﾍﾟｰｽ***
20 LPRINT"FUJITSU";
30 LPRINT CHR$(&H1B);"S";CHR$(&H10);
40 LPRINT"MICRO";
50 LPRINT CHR$(&H1B);"S";CHR$(&H05);
60 LPRINT"7"
```

<印字結果>

```
FUJITSU    MICRO 7
```

(19) 倍幅拡大文字

① ドットプリンタ，シリアルドットプリンタ（MB27401），ビジネスプリンタの場合

<プログラム例>

```
10 REM ***ﾊﾞｲﾊﾊﾞｶｸﾀﾞｲ***
20 LPRINT CHR$(&H1B);CHR$(&H0E);
30 LPRINT"FM-7";
40 LPRINT CHR$(&H14);
50 LPRINT"FM-7"
```

<印字結果>

```
F M - 7FM-7
```

② シリアルドットプリンタ（MB27403）の場合

```
10 REM ***ﾊﾞｲﾊﾊﾞｶｸﾀﾞｲ***
20 LPRINT CHR$(&H1B);"W";CHR$(&H1);
30 LPRINT"FM-7";
40 LPRINT CHR$(&H1B);"W";CHR$(&H0);
50 LPRINT"FM-7"
```

<印字結果>

```
F M - 7FM-7
```

(20) 縮小文字指定

ドットプリンタおよびシリアルドットプリンタ（MB27401）において，縮小文字指定を行なえます．以下に示すものは，すべてSI（シフトイン）機能と同じです．

<プログラム例>

```
10 REM ***ｼｭｸｼｮｳ***
20 LPRINT CHR$(&H1B);CHR$(&H0F);
30 LPRINT"FM-7"
40 LPRINT CHR$(&H12);
50 LPRINT"FM-7"
```

<印字結果>

```
FM-7
FM-7
```

ドットプリンタにおいて，以下のようにも指定できます．

```
10 REM ***ｼｭｸｼｮｳ***
20 LPRINT CHR$(&H1B);"M";
30 LPRINT"FM-7"
40 LPRINT CHR$(&H12);
50 LPRINT"FM-7"
```

<印字結果>

```
FM-7
FM-7
```

(21) プリント開始位置指定

ドットプリンタにおいて，データの印字開始位置を指定できます．

<プログラム例>

```
10 REM ***ﾌﾟﾘﾝﾄ ｶｲｼ ｲﾁ***
20 LPRINT"FUJITSU MICRO 7"
30 LPRINT CHR$(&H10);"0";"5";  ←── スペースを5桁分空けるように指定します．
40 LPRINT"FUJITSU MICRO 7"
50 LPRINT CHR$(&H1B);CHR$(&H10);
60 LPRINT CHR$(&H00);CHR$(&H50);  ←── スペースを80ドット分空けるように指定します．
70 LPRINT"FUJITSU MICRO 7"
```

<印字結果>

```
FUJITSU MICRO 7
     FUJITSU MICRO 7
             FUJITSU MICRO 7
```

(22) インサータのコントロール

ビジネスプリンタは，インサータ（オプション）を使用できます．このインサータにおける単票の吸入，排出が指定できます．

<プログラム例>

```
10 REM ***ｲﾝｻｰﾀ ｺﾝﾄﾛｰﾙ***
20 REM ｷｭｳﾆｭｳ
30 LPRINT CHR$(&H12);"O";
40 LPRINT"FUJITSU MICRO 7"
50 REM ﾊｲｼｭﾂ               ";
60 LPRINT CHR$(&H12);"P";
```

<実行結果>

インサータの単票が吸入された後，データが印字され排出を行ないます．

(23) 紙無検出信号の無効，有効指定

プリンタには，紙が少なくなると，検出器により自動的にOFF-LINE（オフライン）となり，印字が不可能となります．

この場合，プリンタヘッドから検出器まで，用紙が少し残っており，最後まで印字させることはできません．

そこで，検出器を無効とすれば，最後まで印字可能となります．そのときの指定を以下に示します．

・ 無効とする場合

```
LPRINT CHR$(&H1B);"8"
```

・ 有効とする場合

```
LPRINT CHR$(&H1B);"9"
```

14.7 漢字の出力方法

漢字をプリンタへ出力する方法は，以下の方法が考えられます．

(1) 画面へ漢字を表示させ，それをHARDC（ハードコピー）命令により，プリンタへ出力させる．

(2) 画面へ漢字を表示させ，その漢字パターンを，GET@命令により読取り，プリンタへの出力データへ変換する．

(3) 機械語により，漢字パターンをプリンタへの出力データへ変換する．

ここでは，(3)の方法により，F-BASICと機械語サブルーチンを使用する例を示します．機械語サブルーチンは，漢字パターンをプリンタバッファへ送出する働きをします．

プログラム全体の構成は，以下に示す．4つの部分に分けて考えることができます．

① メインプログラム

初期値の設定や，機械語サブルーチンをメモリ上へ展開します．

② 出力する漢字のデータ

DATA（データ）文により，出力する漢字のデータを設定します．

③ 漢字のデータを機械語サブルーチンへ渡す処理

④ 漢字を出力するための機械語サブルーチン

次に，出力方法を示しますが，ビジネスプリンタは，ヘッドピンを16本持っており，他のプリンタとは扱いが異なりますので，分けて説明します．

14.7.1 ドットプリンタおよびシリアルドットプリンタに対する出力方法

(1) 漢字出力するための機械語サブルーチンの作成方法

機械語サブルーチンのメモリダンプリストを以下に示します．このダンプリストを，MON（モン）コマンドにより，メモリへ直接書き，サブルーチンとします．

＜メモリダンプリスト＞

```
       0  1  2  3  4  5 '6  7  8  9  A  B  C  D  E  F        CHARACTER

5A00  81 02 26 1E EC 02 10 83 00 0D 27 17 1E 03 A6 8D     _.&.●..▄..'...ｦ█
5A10  01 23 30 8D 01 20 EF 86 4C 4C A7 8D 01 17 81 78     .#0█. \▄LLｧ█.._x
5A20  24 01 39 30 8D 01 0C AF 8D 01 04 30 8D 00 FE AD     ♦.90█..ｯ█..0█..ｭ
5A30  9F FB FA A6 8D 00 FE 30 8D 00 FB 30 86 AF 8D 01     ｰ..ｦ█..0█..0▄ｯ█.
5A40  6D 30 8D 00 A9 AF 8D 01 67 33 8D 00 E9 11 A3 8D     m0█.ｩｯ█.g3█.♥.｣█
5A50  01 5C 27 36 EC C1 ED 8D 00 92 30 8D 00 90 AF 8D     .¥'6●ﾁ○█.┤0█.┴ｯ█
5A60  00 88 30 8D 00 82 AD 9F FB FA AE 8D 01 42 31 8D     .│0█.▁ｭｰ..ｮ█.B1█
5A70  00 A6 8D 2F 30 01 8D 2B 30 8D 00 98 AF 8D 00 90     .ｦ█/0.█+0█.┌ｯ█.┴
5A80  30 8D 00 8A AD 9F FB FA 20 C3 30 8D 00 70 AC 8D     0█.▌ｭｰ.. ﾃ0█.pｬ█
5A90  01 1E 26 07 8D 33 6F 8D 00 9B 39 AF 8D 01 11 8D     ..&.█3o█.┘9ｯ█..█
5AA0  28 20 A6 86 08 A7 8D 01 09 8D 09 31 21 6A 8D 01     ( ｦ▄.ｧ█..█.1!j█.
5AB0  01 26 F6 39 86 08 A7 8D 00 F9 34 10 68 81 69 A4     .&分9▄.ｧ█..4.h_i､
5AC0  6A 8D 00 EF 26 F6 35 10 39 CC 00 05 ED 8D 00 42     j█.\&分5.9ﾌ..○█.B
5AD0  30 8D 00 54 AF 8D 00 38 30 8D 00 32 AD 9F FB FA     0█.Tｯ█.80█.2ｭｰ..
5AE0  CC 00 14 ED 8D 00 2B 39 16 00 00 00 00 00 00 00     ﾌ..○█.+9........
5AF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
```

```
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F        CHARACTER

5B00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E 00     ................
5B10  00 00 00 14 1B 4C 10 00 00 00 00 00 00 00 00 00     .....L..........
5B20  00 00 00 00 00 00 00 00 1B 41 08 0D 0A 0E 00 00     .........A......
5B30  00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5B40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5B50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5B60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5B70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5B80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5B90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5BA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
5BB0  00 00 00 00 00 37 4A 26 F1 DE 2D A6 00 16 DB 25     .....7J&円ﾞ-ｦ..ﾛ%
5BC0  D7 25 4D 27 13 DE 2F 09 E6 00 D8 2A 37 09 E6 00     ﾗ%M'.ﾞ/.◥.ﾘ*7.◥.
5BD0  37 09 E6 00 37 4A 26 EF D6 25 37 DE 31 09 09 DF     7.◥.7J&\ﾖ%7ﾞ1..ﾟ
5BE0  25 DE 2D 09 09 DF 27 DE 2F A6 03 97 31 DE 2B A6     %ﾞ-..ﾟ'ﾞ/ｦ.┃1ﾞ+ｦ
5BF0  02 97 32 5F 96 00 00 00 00 00 00 00 00 00 00 00     .┃2_│...........
```

① MON⏎と入力します．

② 画面へ“＊”が表示されますので，M5A00⏎と入力します．

③ “5A00 00－”と表示されます．

これは，メモリの番地である“5A00”と，その番地の内容である“00”を出力しているのです．

④ 次に，メモリダンプリストの内容を入力していきます．

5A00 00－81

⑤ 次の番地の内容が同様に表示されますので，次々と入力していきます．

＜入力例＞

```
Ready
MON

*M5A00
5A00 00-81
5A01 00-02
5A02 00-26
5A03 00-1E
5A04 00-EC
5A05 00-02
5A06 00-10
5A07 00-
```

⑥ 入力が終了すれば，カセット（またはフロッピィディスク）へセーブします．

SAVEM “CAS0：ファイル名”；&H5A00，&H5AEF，&H5A00

（2） プログラム例とその説明

＜プログラム例＞

```
10 REM***ｶﾝｼﾞ ｼｭﾂﾘｮｸ***
20 CLEAR 300,&H59FF
30 LOADM"CAS0:KANJI"
40 DEF USR1=&H5A00
50 GOSUB 100
60 END
70 '
80 '
90 '
```

（行10～60：メインプログラム）

```
100 REM***ｻﾌﾞﾙｰﾁﾝ***
110 READ A%:N=N+1
120    IF A%=&H00  THEN 200
130    IF A%<>&H0D THEN 170
140      A%=&H0D:N=0
150      A%=USR1(A%):A%=USR1(A%)
160     GO TO 110
170    IF N=60 THEN 140
180 I%=USR1(A%)
190     GO TO 110
200 A%=&H0D
210 A%=USR1(A%):A%=USR1(A%)
220 RETURN
```

漢字のデータを機械語サブルーチンへ渡す処理

```
230 '
240 '
250 '
1000 REM***ﾃﾞｰﾀ***
1010 DATA &H2433,&H244E,&H2557,&H256D
1020 '       ｺ      ﾉ      ﾌﾟ      ﾛ
1030 DATA &H2530,&H2569,&H2560,&H4E63
1040 '       ｸﾞ      ﾗ      ﾑ      ﾚｲ
1050 DATA &H244F,&H2122,&H2557,&H256A
1060 '       ﾊ      ､      ﾌﾟ      ﾘ
1070 DATA &H2573,&H253F,&H2558,&H2122
1080 '       ﾝ      ﾀ      ﾍ      ､
1090 DATA &H3441,&H3B7A,&H2472,&H3D50
1100 '       ｶﾝ     ｼﾞ      ﾖ      ｼｭﾂ
1110 DATA &H4E4F,&H2439,&H246B,&H243F
1120 '       ﾘｮｸ    ｽ      ﾙ      ﾀ
1130 DATA &H2461,&H244E,&H2462,&H244E
1140 '       ﾒ      ﾉ      ﾓ      ﾉ
1150 DATA &H2447,&H2439,&H2123,&H0D
1160 '       ﾃﾞ      ｽ      ｡
1170 DATA &H2346,&H215D,&H2342,&H2341
1180 '       F      -      B      A
1190 DATA &H2353,&H2349,&H2343,&H2448
1200 '       S      I      C      ﾄ
1210 DATA &H2122,&H3521,&H3323,&H386C
1220 '       ､      ｷ      ｶｲ     ｺﾞ
1230 DATA &H244B,&H2468,&H246A,&H2122
1240 '       ﾆ      ﾖ      ﾘ      ､
1250 DATA &H393D,&H402E,&H2435,&H246C
1260 '       ｺｳ     ｾｲ     ｻ      ﾚ
1270 DATA &H245E,&H2439,&H2123,&H00
1280 '       ﾏ      ｽ      ｡
```

出力する漢字のデータ

＜プログラム説明＞

・行番号 20

機械語サブルーチンをロードするために，F-BASIC の使用する領域を制限しています．

・行番号 30

機械語サブルーチンをメモリへロードします．機械語サブルーチンをセーブする際，格納番地(&H5A00) を指定すれば，オフセット値を与える必要はありません．

・行番号 40

機械語サブルーチンの開始番地を指定します．

・行番号 50

処理をサブルーチンへ渡します．

・行番号 60

プログラムの終了を示します．

・行番号 110 （行番号100以降は，サブルーチンです）

出力する漢字のコードを DATA 文より読取ります．

・行番号 120

DATA 文におけるコードが &H00 であれば，行番号 200 へ処理を移し，プリンタバッファ内のデータを印字後，メインプログラムへもどります．

・行番号 130

DATA 文におけるコードが &H0D でなければ，行番号 170 へ処理を移し，プリンタバッファへデータを蓄えます．

・行番号 140

行番号 130 より，コードが &H0D であれば処理がこの行番号へ移ります．

・行番号 150，160

プリンタバッファへ改行コード（&H0D) を送出して，バッファ内のデータの印字を行ないます．その後，さらにデータを読取るため，行番号 160 により行番号 110 へ実行を移します．

・行番号 170

プリンタへの印字桁数を 60 桁と指定して，60 桁分のデータが，プリンタバッファへ蓄えられると，行番号 140 へ実行を移し，改行コードをプリンタへ送出して，印字を行ないます．

・行番号 1000 以降

すべて出力するための漢字データです．

<印字結果>

このプログラム例は、プリンタへ、漢字を出力するためのものです。

F－BASICと、機械語により、構成されます。

14.7.2　ビジネスプリンタに対する出力方法

ビジネスプリンタも，14.7.1項において示す方法と同じです．メモリダンプリストを以下に示します．

<メモリダンプリスト>

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	CHARACTER
5000	81	02	26	0E	EC	02	10	83	00	0D	27	6A	8D	05	8D	16	_.&.●..▬..'J■.■.
5010	8D	3F	39	ED	8D	00	7A	30	8D	00	78	AF	8D	00	70	30	■?9○■.z0■.xｯ■.p0
5020	1A	AD	9F	FB	FA	39	30	8D	00	69	31	8D	00	89	8D	04	.ｭ/..90■.i1■.▌■.
5030	30	01	31	A5	5F	4F	68	86	66	A5	8B	02	81	10	26	F6	0.1･_Oh■f･▌._.&ﾀ
5040	5C	68	86	66	A5	8B	02	81	20	26	F6	5C	C1	10	26	E5	¥h■f･▌._ &ﾀ¥ﾁ.&◣
5050	39	CC	1B	34	ED	8D	00	5B	CC	00	10	ED	8D	00	56	30	9ﾌ.4○■.[ﾌ..○■.V0
5060	8D	00	50	AF	8D	00	0B	30	8D	00	05	AD	9F	FB	FA	39	■.Pｯ■..0■..ｭ/..9
5070	0E	00	00	00	00	24	30	8D	00	11	AF	8D	00	09	30	1A	.....$0■..ｯ■..0.
5080	AD	9F	FB	FA	39	0E	00	00	00	00	02	0D	0A	16	00	00	ｭ/..9...........
5090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
50A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
50B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
50C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
50D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
50E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
50F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....

以下に，漢字を出力するためのプログラム例を示します．

<プログラム例>

```
100 ' ﾋﾞｼﾞﾈｽ ﾌﾟﾘﾝﾀ ﾍﾉｶﾝｼﾞ ｼｭﾂﾘｮｸ
110 CLEAR 300,&H4FFF
120 LOADM"CAS0:B-KANJI"
130 DEF USR1=&H5000
140 FOR I=1 TO 5
150   READ A%
160   A%=USR1(A%)
170 NEXT I
180 A%=&H0D
190 A%=USR1(A%)
200 END
1000 DATA &H3441,&H3B7A,&H3D50,&H4E4F
1010 DATA &H4E63
```

<プログラム説明>

・行番号 110

CLEAR 命令により，BASIC プログラムの使用領域を制限します．

・行番号 120

機械語のサブルーチンをメモリ上へ展開します．ここでのファイル名は，“B-KANJI” としています．

・行番号 140

機械語サブルーチンの実行開始番地を指定します．

・行番号 150～200

漢字を出力するためのデータをプリンタへ送出しています．

・行番号 220

漢字のデータです．

<印字結果>

漢字出力例

第15章　回　線　制　御

15.1 概　　要

RS-232Cとは，本来，モデムとデータ端末装置との接続に関する規格として，EIA (Electronic Industries Association) により決められたインタフェースです．

現在は，モデムに限らず，プロッタやプリンタ等の周辺装置に，RS-232C準拠という形により広く利用されています．

FM-7においては，オプションのRS-232Cインタフェースカードを，本体のオプションスロットへ装着することにより，RS-232Cインタフェースを持つ周辺装置と接続することができます．

また，RS-232Cインタフェースカード上のディップスイッチの設定により，ボーレートを300～19,200ボーに指定できます．

次ページに，RS-232Cインタフェースカードのコネクタ番号と信号線名の対応表を示します．

15.2　インタフェース

カードコネクタ番号	信　　号　　名	RS-232C コネクタ番号
1		—
2		13
3		25
4		12
5	ST1	24
6		11
7		23
8		10
9		22
10		9
11		21
12	CARRIER DETECT	8
13	DATA TERMINAL READY	20
14	SIGNAL GND	7
15		19
16	DATA SET READY	6
17		18
18	CLEAR TO SEND	5
19	RECEIVE TIMING	17
20	REQUEST TO SEND	4
21		16
22	RECEIVED DATA	3
23	ST2	15
24	TRANSMITTED DATA	2
25		14
26	FRAME GND	1

※RS-232C仕様準拠，フラットケーブルの標識線（青色）を1としております．

・CARRIER DETECT（キャリア ディテクト）

音響カプラ等のキャリアを検出するための信号です．

・DATA TERMINAL READY（データ ターミナル レディー）

本体が，送受信可能であることを示す信号です．

・DATA SET READY (データ セット レディー)

周辺装置が動作可能であることを示す信号です.

・CLEAR TO SEND (クリア ツゥ センド)

周辺装置が, データを送受信可能であることを示す信号です.

・REQUEST TO SEND (リクエスト ツゥ センド)

周辺装置に対し, データの送信要求信号です.

・RECEIVED DATA (レシーブ データ)

データの入力信号です.

・TRANSMITTED DATA (トランスミット データ)

データの出力信号です.

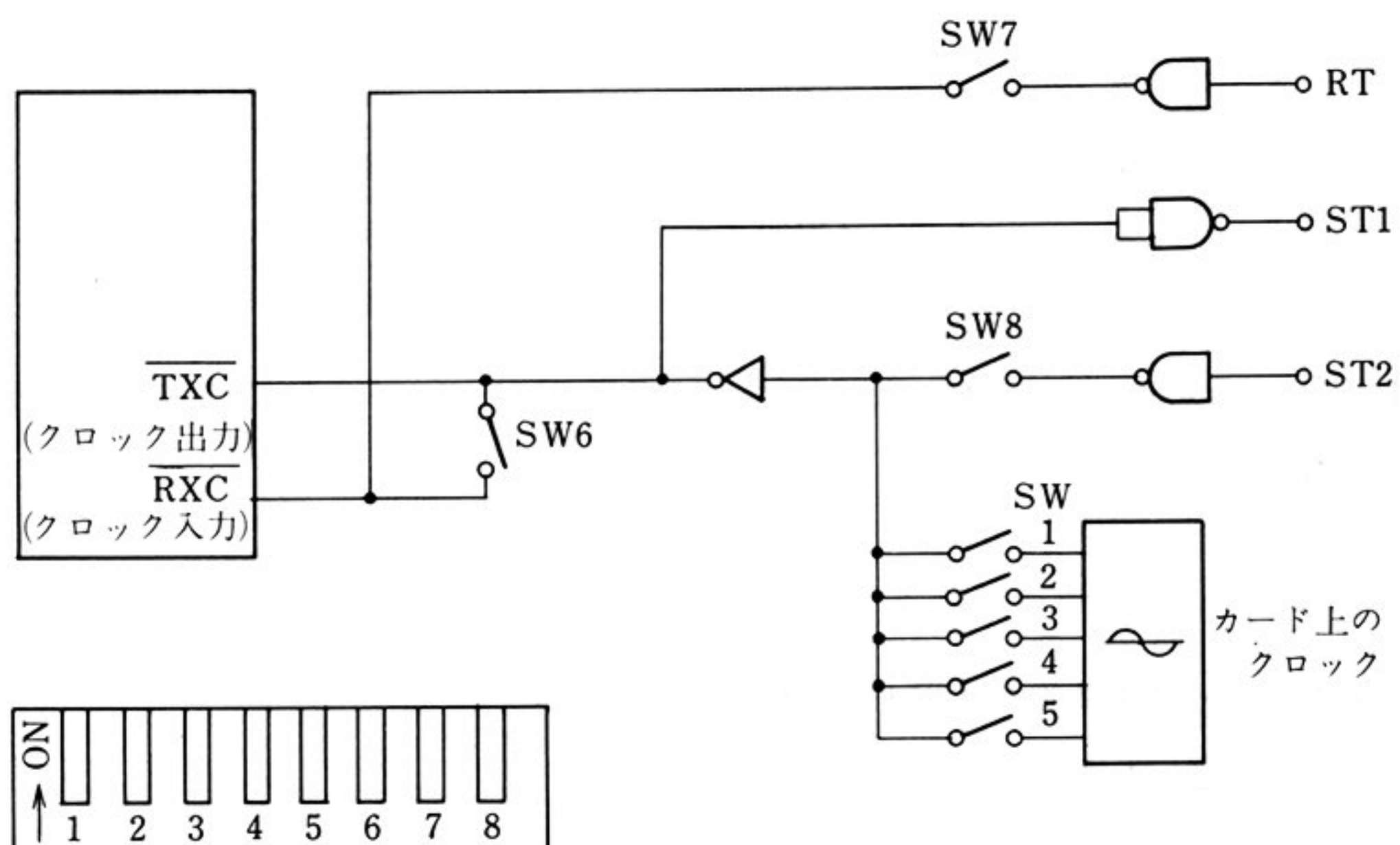
・RECEIVE TIMING (RT), ST1, ST2

RS-232C インタフェースカード上における, データ送受信用のクロックを使用しない場合, これらの信号線より, クロックの入力を行ないます.

・SIGNAL GND, FRAME GND

グランド端子です.

15.3 ディップスイッチの設定方法

以下に, ディップ スイッチに関する回路図を示し, 設定の意味を示します.

ディップスイッチの番号は, SW1〜8の番号と対応しています

<table>
<tr><th rowspan="2">使　用　方　法</th><th rowspan="2">使用するクロック</th><th colspan="4">ディップスイッチ番号</th></tr>
<tr><th>1～5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td rowspan="3">送受信共に，同一のクロックを使用する場合</td><td>カード上のクロック</td><td>ボーレート選択</td><td>○</td><td>×</td><td>×</td></tr>
<tr><td>ST2端子より入力</td><td>×</td><td>○</td><td>×</td><td>○</td></tr>
<tr><td>RT端子より入力</td><td>×</td><td>○</td><td>○</td><td>×</td></tr>
<tr><td rowspan="2">送信・受信クロックを分離して使用する場合</td><td>カード上のクロックを送信とし，RT端子よりクロックを入力する</td><td>ボーレート選択</td><td>×</td><td>○</td><td>×</td></tr>
<tr><td>ST2端子より，送信用のクロックを入力して，受信用のクロックをRT端子より入力</td><td>×</td><td>×</td><td>○</td><td>○</td></tr>
</table>

※○はON，×はOFFを示します．送信用クロックは，常にST1端子より出力されます．

15.4　ボーレートの選択

ボーレートとは，データを転送する速さの単位です．1秒間に送受信可能なビット数で表わし，1秒間に300ビット送受信可能であれば，300ボーとなります．

RS-232Cインタフェースカードにおいては，ボーレートをカード上のディップスイッチにより設定します．以下に設定方法を示します．

ディップスイッチ番号	転送モード	
	Slow (1/64分周) モード	Fast (1/16分周) モード
1	300ボー	1200ボー
2	600ボー	2400ボー
3	1200ボー	4800ボー
4	2400ボー	9600ボー
5	4800ボー	19200ボー

Slow/Fastモードの選択は，TERMコマンドや，RS-232Cポートに対するOPEN命令のオプションにより指定します．

<1200ボーを設定する方法>

(1)　ディップスイッチの3番を使用する場合

Slowモードを指定します．

OPEN "O", #1, "COM0 : (S 8 N 2)"

- "O"：アウトプット ファイルを示します
- #1：ファイル番号
- COM0：デバイス名
- 2 → データにストップビットを2ビット付けるよう指定しています.
- N → パリティビットを使用しないことを示します.
- 8 → データは, 8ビット構成であることを指定します.
- S → Slow モードを指定します.

（2） ディップスイッチの1番を使用する場合

Fast モードを指定し, その他のオプション機能は同一とします.

OPEN "O", #1, "COM0 : (F8N2)"

- F → Fast モードを指定します.

15.5 F-BASIC におけるエラー発生表示

F-BASIC のターミナルモードで使用している時, および, 回線制御機能を用いてデータの送受信を行なっている時の, データ受信時に発生するエラーは, 次のものが考えられます.

・**フレーミングエラー**

データのストップビットを検出できなかった時に発生します（回線障害）

・**パリティエラー**

データへパリティビットを指定した時, 正しいパリティを検出できなかった時に発生します（回線障害）.

・**オーバランエラー**

データ受信時において, 取り込まれたシリアルデータはパラレルデータに変換され, メモリへ展開されますが, この処理において, メモリへ展開し終らないうちに, 次のシリアルデータをパラレルデータに変換しようとした時に発生します（TERM コマンド使用時においては発生しません）.

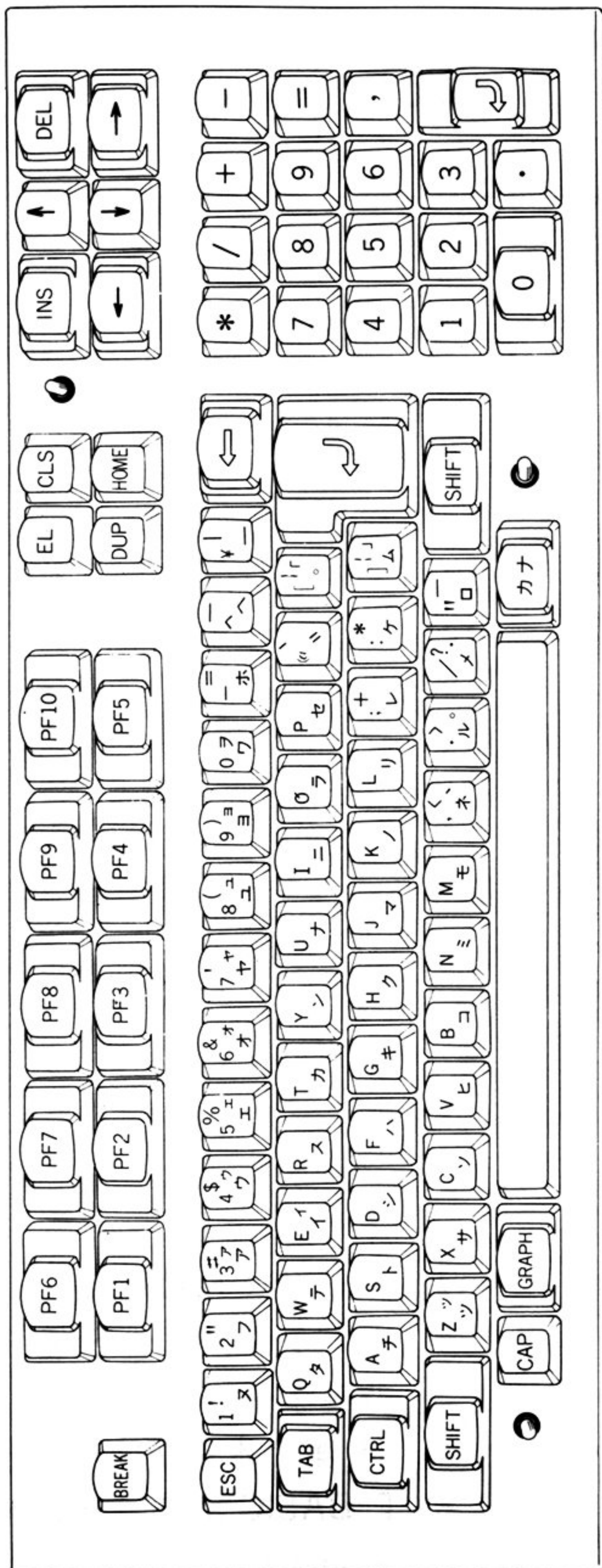
・**バッファオーバフロー**

メモリには, データをたくわえるバッファ（127文字分）を持っていますが, ボーレートの選択によって, データ転送のスピードが早すぎたときなど, バッファ内のデータを処理し切れず, データがバッファからあふれる（オーバフローする）と発生します.

以上のエラーに対する表示方法を以下に示します.

エラー名	処　理　内　容
フレーミングエラーまたはパリティエラー	エラーが発生したデータを強制的に疑問符（?）に変更し, 受信動作を継続します.
オーバランエラー	エラー (Device I/O Error) となり, 処理を終了します.
バッファオーバフロー	エラー (Buffer Overflow) となり, 処理を終了します.

付　　　録

1　キーボード配列図
2　キャラクタコード表
3　非漢字一覧表
4　JIS 第 1 水準漢字一覧表
5　F-BASIC エラーリスト
6　機器接続図

付録 1　キーボード配列図

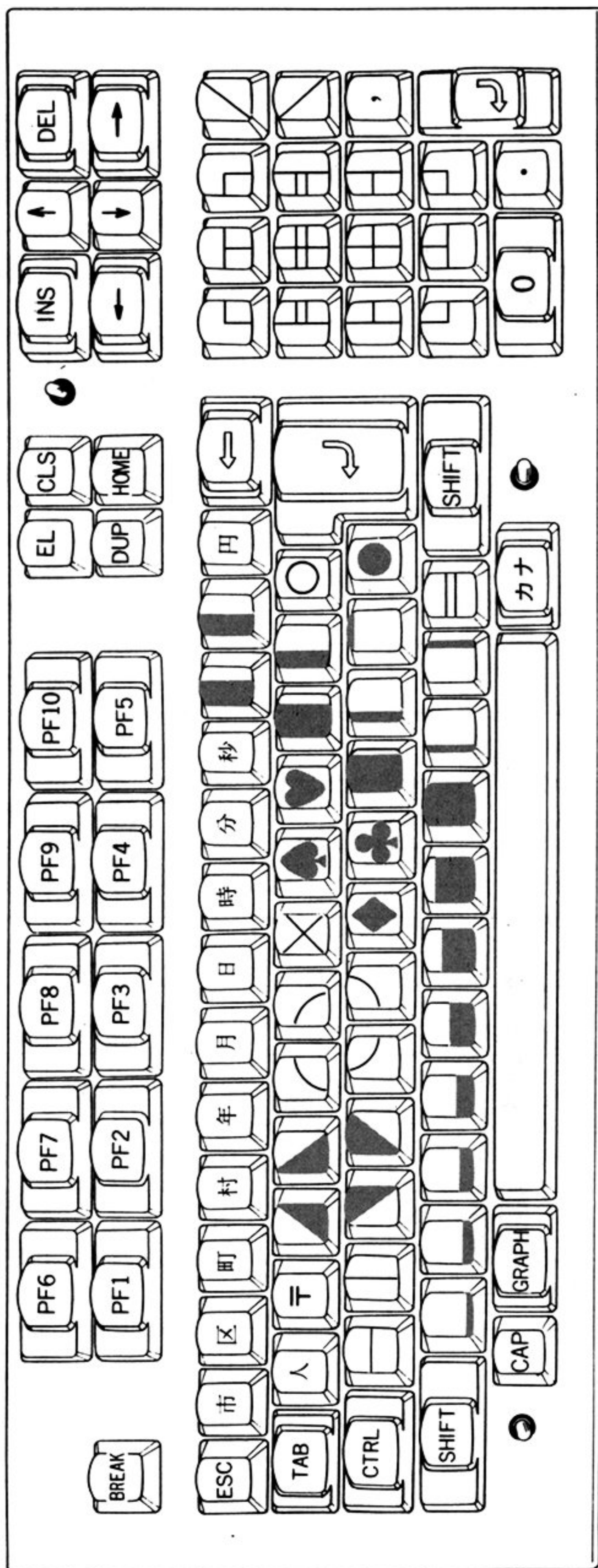
［キーボードの説明］

FM-7 のキーボードは JIS 配列に準拠し，専用マイクロコンピュータ（4 ビットマイクロコンピュータ・MB88401）により制御されており，強力なエディット機能を備えています．

また，リピート機能により，同一キーを約 0.7 秒以上，押し続けますと，同じ文字が連続して表示されます．

更に，状態表示機能として，英大文字／カナ／挿入の各モード LED 表示により，一層使いやすくなっています．

次に，通常モードのキー配列，グラフィックモードのキー配列，キャラクタコード表を示します．

通常モードのキー配列

グラフィックモードのキー配列

付録2　キャラクタコード表

下位＼上位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DE	(space)	0	@	P	`	p	▁	┴		ー	タ	ミ	═	╳
1	SH	D1	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	╞	円
2	SX	D2	"	2	B	R	b	r	▃	┤	「	イ	ツ	メ	╪	年
3	EX	D3	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	╡	月
4	ET	D4	$	4	D	T	d	t	▅	▔	、	エ	ト	ヤ	◢	日
5	EQ	NK	%	5	E	U	e	u	▆	─	・	オ	ナ	ユ	◣	時
6	AK	SN	&	6	F	V	f	v	▇	│	ヲ	カ	ニ	ヨ	◥	分
7	BL	EB	'	7	G	W	g	w	█	▕	ァ	キ	ヌ	ラ	◤	秒
8	BS	CN	(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ	♠	〒
9	HT	EM	)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	市
A	LF	SB	*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	♦	区
B	HM	EC	+	;	K	[	k	{	▌	┘	ォ	サ	ヒ	ロ	♣	町
C	CL	→	,	<	L	¥	l	|	▋	╭	ャ	シ	フ	ワ	●	村
D	CR	←	-	=	M	]	m	}	▊	╮	ュ	ス	ヘ	ン	○	人
E	SO	↑	.	>	N	^	n	￣	▉	╰	ョ	セ	ホ	゛	╱	▒
F	SI	↓	/	?	O	_	o	DL	┼	╯	ッ	ソ	マ	゜	╲	

付録３　非漢字一覧表

コードはすべて 16 進形式

記号

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
212X			、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨
213X	＾	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／
214X	＼	～	∥	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］
215X	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×
216X	÷	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥
217X	＄	¢	£	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇	
222X		◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	〓	

（例えば，％のコードは2173と読みます．実際の使用には“&H”をつけて，「&H 2173」とします）

英・数字

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
233X	０	１	２	３	４	５	６	７	８	９						
234X		Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ
235X	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ					
236X		ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ
237X	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ					

ひらがな

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
242X		ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
243X	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
244X	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
245X	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
246X	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
247X	ゐ	ゑ	を	ん												

カタカナ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
252X		ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク
253X	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
254X	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ
255X	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ
256X	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
257X	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ									

(例えば，ケのコードは2531と読みます．実際の使用には"&H"をつけて，「&H2531」とします．)

ギリシャ文字

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
262X		Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
263X	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω							
264X		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
265X	π	ρ	σ	τ	υ	φ	χ	ψ	ω							

ロシア文字

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
272X		А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
273X	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
274X	Ю	Я														
275X		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
276X	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
277X	ю	я														

付録4 JIS第1水準漢字一覧表

コードはすべて16進形式

ア

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
302X		亜	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥
303X	旭	葦	芦	鯵	梓	圧	斡	扱	宛	姐	虻	飴	絢	綾	鮎	或
304X	粟	袷	安	庵	按	暗	案	闇	鞍	杏						

（例えば，安のコードは3042と読みます．実際の使用には“&H”をつけて，「&H3042」とします）

イ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
304X											以	伊	位	依	偉	囲
305X	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃
306X	萎	衣	謂	違	遺	医	井	亥	域	育	郁	磯	一	壱	溢	逸
307X	稲	茨	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭	
312X		院	陰	隠	韻	吋										

ウ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
312X							右	宇	烏	羽	迂	雨	卯	鵜	窺	丑
313X	碓	臼	渦	嘘	唄	欝	蔚	鰻	姥	厩	浦	瓜	閏	噂	云	運
314X	雲															

エ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
314X		荏	餌	叡	営	嬰	影	映	曳	栄	永	泳	洩	瑛	盈	穎
315X	頴	英	衛	詠	鋭	液	疫	益	駅	悦	謁	越	閲	榎	厭	円
316X	園	堰	奄	宴	延	怨	掩	援	沿	演	炎	焔	煙	燕	猿	縁
317X	艶	苑	薗	遠	鉛	鴛	塩									

オ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
317X								於	汚	甥	凹	央	奥	往	応	
322X		押	旺	横	欧	殴	王	翁	襖	鴬	鴎	黄	岡	沖	荻	億
323X	屋	憶	臆	桶	牡	乙	俺	卸	恩	温	穏	音				

カ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
323X													下	化	仮	何

次頁につづく

コードはすべて16進形式

ク

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
366X						九	倶	句	区	狗	玖	矩	苦	軀	駆	駈
367X	駒	具	愚	虞	喰	空	偶	寓	遇	隅	串	櫛	釧	屑	屈	
372X		掘	窟	沓	靴	轡	窪	熊	隈	粂	栗	繰	桑	鍬	勲	君
373X	薫	訓	群	軍	郡											

（例えば，君のコードは 372F と読みます．実際の使用は "&H" をつけて「&H327F」とします．）

ケ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
373X						卦	袈	祁	係	傾	刑	兄	啓	圭	珪	型
374X	契	形	径	恵	慶	慧	憩	掲	携	敬	景	桂	渓	畦	稽	系
375X	経	継	繋	罫	茎	荊	蛍	計	詣	警	軽	頚	鶏	芸	迎	鯨
376X	劇	戟	撃	激	隙	桁	傑	欠	決	潔	穴	結	血	訣	月	件
377X	倹	倦	健	兼	券	剣	喧	圏	堅	嫌	建	憲	懸	拳	捲	
382X		検	権	牽	犬	献	研	硯	絹	県	肩	見	謙	賢	軒	遣
383X	鍵	険	顕	験	鹸	元	原	厳	幻	弦	減	源	玄	現	絃	舷
384X	言	諺	限													

コ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
384X				乎	個	古	呼	固	姑	孤	己	庫	弧	戸	故	枯
385X	湖	狐	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇	顧	鼓	五	互
386X	伍	午	呉	吾	娯	後	御	悟	梧	檎	瑚	碁	語	誤	護	醐
387X	乞	鯉	交	佼	侯	候	倖	光	公	功	効	勾	厚	口	向	
392X		后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	広	庚	康
393X	弘	恒	慌	抗	拘	控	攻	昂	晃	更	杭	校	梗	構	江	洪
394X	浩	港	溝	甲	皇	硬	稿	糠	紅	紘	絞	綱	耕	考	肯	肱
395X	腔	膏	航	荒	行	衡	講	貢	購	郊	酵	鉱	砿	鋼	閤	降
396X	項	香	高	鴻	剛	劫	号	合	壕	拷	濠	豪	轟	麹	克	刻
397X	告	国	穀	酷	鵠	黒	獄	漉	腰	甑	忽	惚	骨	狛	込	
3A2X		此	頃	今	困	坤	墾	婚	恨	懇	昏	昆	根	梱	混	痕
3A3X	紺	艮	魂													

コードはすべて16進形式

カ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	324X	伽	価	佳	加	可	嘉	夏	嫁	家	寡	科	暇	果	架	歌	河
	325X	火	珂	禍	禾	稼	箇	花	苛	茄	荷	華	菓	蝦	課	嘩	貨
	326X	迦	過	霞	蚊	俄	峨	我	牙	画	臥	芽	蛾	賀	雅	餓	駕
	327X	介	会	解	回	塊	壊	廻	快	怪	悔	恢	懐	戒	拐	改	
	332X		魁	晦	械	海	灰	界	皆	絵	芥	蟹	開	階	貝	凱	劾
	333X	外	咳	害	崖	慨	概	涯	碍	蓋	街	該	鎧	骸	浬	馨	蛙
	334X	垣	柿	蠣	鈎	劃	嚇	各	廓	拡	撹	格	核	殻	獲	確	穫
	335X	覚	角	赫	較	郭	閣	隔	革	学	岳	楽	額	顎	掛	笠	樫
	336X	橿	梶	鰍	潟	割	喝	恰	括	活	渇	滑	葛	褐	轄	且	鰹
	337X	叶	椛	樺	鞄	株	兜	竃	蒲	釜	鎌	噛	鴨	栢	茅	萱	
	342X		粥	刈	苅	瓦	乾	侃	冠	寒	刊	勘	勧	巻	喚	堪	姦
	343X	完	官	寛	干	幹	患	感	慣	憾	換	敢	柑	桓	棺	款	歓
	344X	汗	漢	澗	潅	環	甘	監	看	竿	管	簡	緩	缶	翰	肝	艦
	345X	莞	観	諌	貫	還	鑑	間	閑	関	陥	韓	館	舘	丸	含	岸
	346X	巌	玩	癌	眼	岩	翫	贋	雁	頑	顔	願					

(例えば，可のコードは3244と読みます．実際の使用には"&H"をつけて「&H3244」とします．)

キ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	346X												企	伎	危	喜	器
	347X	基	奇	嬉	寄	岐	希	幾	忌	揮	机	旗	既	期	棋	棄	
	352X		機	帰	毅	気	汽	畿	祈	季	稀	紀	徽	規	記	貴	起
	353X	軌	輝	飢	騎	鬼	亀	偽	儀	妓	宜	戯	技	擬	欺	犠	疑
	354X	祇	義	蟻	誼	議	掬	菊	鞠	吉	吃	喫	桔	橘	詰	砧	杵
	355X	黍	却	客	脚	虐	逆	丘	久	仇	休	及	吸	宮	弓	急	救
	356X	朽	求	汲	泣	灸	球	究	窮	笈	級	糾	給	旧	牛	去	居
	357X	巨	拒	拠	挙	渠	虚	許	距	鋸	漁	禦	魚	亨	享	京	
	362X		供	侠	僑	兇	競	共	凶	協	匡	卿	叫	喬	境	峡	強
	363X	彊	怯	恐	恭	挟	教	橋	況	狂	狭	矯	胸	脅	興	蕎	郷
	364X	鏡	響	饗	驚	仰	凝	尭	暁	業	局	曲	極	玉	桐	粁	僅
	365X	勤	均	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊	芹	菌	衿	襟
	366X	謹	近	金	吟	銀											

サ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3A3X				些	佐	叉	唆	嵯	左	差	査	沙	瑳	砂	詐	鎖
3A4X	裟	坐	座	挫	債	催	再	最	哉	塞	妻	宰	彩	才	採	栽
3A5X	歳	済	災	采	犀	砕	砦	祭	斎	細	菜	裁	載	際	剤	在
3A6X	材	罪	財	冴	坂	阪	堺	榊	肴	咲	崎	埼	碕	鷺	作	削
3A7X	咋	搾	昨	朔	柵	窄	策	索	錯	桜	鮭	笹	匙	冊	刷	
3B2X		察	拶	撮	擦	札	殺	薩	雑	皐	鯖	捌	錆	鮫	皿	晒
3B3X	三	傘	参	山	惨	撒	散	桟	燦	珊	産	算	纂	蚕	讃	賛
3B4X	酸	餐	斬	暫	残											

(例えば，山のコードは3B33と読みます．実際の使用は“&H”をつけて「&H3B33」とします)

シ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3B4X						仕	仔	伺	使	刺	司	史	嗣	四	士	始
3B5X	姉	姿	子	屍	市	師	志	思	指	支	孜	斯	施	旨	枝	止
3B6X	死	氏	獅	祉	私	糸	紙	紫	肢	脂	至	視	詞	詩	試	誌
3B7X	諮	資	賜	雌	飼	歯	事	似	侍	児	字	寺	慈	持	時	
3C2X		次	滋	治	爾	璽	痔	磁	示	而	耳	自	蒔	辞	汐	鹿
3C3X	式	識	鴫	竺	軸	宍	雫	七	叱	執	失	嫉	室	悉	湿	漆
3C4X	疾	質	実	蔀	篠	偲	柴	芝	屡	蕊	縞	舎	写	射	捨	赦
3C5X	斜	煮	社	紗	者	謝	車	遮	蛇	邪	借	勺	尺	杓	灼	爵
3C6X	酌	釈	錫	若	寂	弱	惹	主	取	守	手	朱	殊	狩	珠	種
3C7X	腫	趣	酒	首	儒	受	呪	寿	授	樹	綬	需	囚	収	周	
3D2X		宗	就	州	修	愁	拾	洲	秀	秋	終	繍	習	臭	舟	蒐
3D3X	衆	襲	讐	蹴	輯	週	酋	酬	集	醜	什	住	充	十	従	戎
3D4X	柔	汁	渋	獣	縦	重	銃	叔	夙	宿	淑	祝	縮	粛	塾	熟
3D5X	出	術	述	俊	峻	春	瞬	竣	舜	駿	准	循	旬	楯	殉	淳
3D6X	準	潤	盾	純	巡	遵	醇	順	処	初	所	暑	曙	渚	庶	緒
3D7X	署	書	薯	藷	諸	助	叙	女	序	徐	恕	鋤	除	傷	償	
3E2X		勝	匠	升	召	哨	商	唱	嘗	奨	妾	娼	宵	将	小	少
3E3X	尚	庄	床	廠	彰	承	抄	招	掌	捷	昇	昌	昭	晶	松	梢
3E4X	樟	樵	沼	消	渉	湘	焼	焦	照	症	省	硝	礁	祥	称	章
3E5X	笑	粧	紹	肖	菖	蒋	蕉	衝	裳	訟	証	詔	詳	象	賞	醤

次頁につづく

コードはすべて16進形式

シ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3E6X	鉦	鍾	鐘	障	鞘	上	丈	丞	乗	冗	剰	城	場	壌	嬢	常
3E7X	情	擾	条	杖	浄	状	畳	穣	蒸	譲	醸	錠	嘱	埴	飾	
3F2X		拭	植	殖	燭	織	職	色	触	食	蝕	辱	尻	伸	信	侵
3F3X	唇	娠	寝	審	心	慎	振	新	晋	森	榛	浸	深	申	疹	真
3F4X	神	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	刃
3F5X	塵	壬	尋	甚	尽	腎	訊	迅	陣	靭						

(例えば，上のコードは3E65と読みます．実際の使用は"&H"をつけて「&H3E65」とします)

ス

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3F5X											笥	諏	須	酢	図	厨
3F6X	逗	吹	垂	帥	推	水	炊	睡	粋	翠	衰	遂	酔	錐	錘	随
3F7X	瑞	髄	崇	嵩	数	枢	趨	雛	据	杉	椙	菅	頗	雀	裾	
402X		澄	摺	寸												

セ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
402X					世	瀬	畝	是	凄	制	勢	姓	征	性	成	政
403X	整	星	晴	棲	栖	正	清	牲	生	盛	精	聖	声	製	西	誠
404X	誓	請	逝	醒	青	静	斉	税	脆	隻	席	惜	戚	斥	昔	析
405X	石	積	籍	績	脊	責	赤	跡	蹟	碩	切	拙	接	摂	折	設
406X	窃	節	説	雪	絶	舌	蝉	仙	先	千	占	宣	専	尖	川	戦
407X	扇	撰	栓	栴	泉	浅	洗	染	潜	煎	煽	旋	穿	箭	線	
412X		繊	羨	腺	舛	船	薦	詮	賎	践	選	遷	銭	銑	閃	鮮
413X	前	善	漸	然	全	禅	繕	膳	糎							

ソ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
413X										噌	塑	岨	措	曾	曽	楚
414X	狙	疏	疎	礎	祖	租	粗	素	組	蘇	訴	阻	遡	鼠	僧	創
415X	双	叢	倉	喪	壮	奏	爽	宋	層	匝	惣	想	捜	掃	挿	掻
416X	操	早	曹	巣	槍	槽	漕	燥	争	痩	相	窓	糟	総	綜	聡
417X	草	荘	葬	蒼	藻	装	走	送	遭	鎗	霜	騒	像	増	憎	
422X		臓	蔵	贈	造	促	側	則	即	息	捉	束	測	足	速	俗

次頁につづく

ソ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	423X	属	賊	族	続	卒	袖	其	揃	存	孫	尊	損	村	遜		

(例えば，存のコードは 4238 と読みます．実際の使用は “&H” をつけて「&H4238」とします)

タ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	423X															他	多
	424X	太	汰	詑	唾	堕	妥	惰	打	柁	舵	楕	陀	駄	騨	体	堆
	425X	対	耐	岱	帯	待	怠	態	戴	替	泰	滞	胎	腿	苔	袋	貸
	426X	退	逮	隊	黛	鯛	代	台	大	第	醍	題	鷹	滝	瀧	卓	啄
	427X	宅	托	択	拓	沢	濯	琢	託	鐸	濁	諾	茸	凧	蛸	只	
	432X		叩	但	達	辰	奪	脱	巽	竪	辿	棚	谷	狸	鱈	樽	誰
	433X	丹	単	嘆	坦	担	探	旦	歎	淡	湛	炭	短	端	箪	綻	耽
	434X	胆	蛋	誕	鍛	団	壇	弾	断	暖	檀	段	男	談			

チ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	434X														値	知	地
	435X	弛	恥	智	池	痴	稚	置	致	蜘	遅	馳	築	畜	竹	筑	蓄
	436X	逐	秩	窒	茶	嫡	着	中	仲	宙	忠	抽	昼	柱	注	虫	衷
	437X	註	酎	鋳	駐	樗	瀦	猪	苧	著	貯	丁	兆	凋	喋	寵	
	442X		帖	帳	庁	弔	張	彫	徴	懲	挑	暢	朝	潮	牒	町	眺
	443X	聴	脹	腸	蝶	調	諜	超	跳	銚	長	頂	鳥	勅	捗	直	朕
	444X	沈	珍	賃	鎮	陳											

ツ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	444X						津	墜	椎	槌	追	鎚	痛	通	塚	栂	摑
	445X	槻	佃	漬	柘	辻	蔦	綴	鍔	椿	潰	坪	壺	嬬	紬	爪	吊
	446X	釣	鶴														

テ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	446X			亭	低	停	偵	剃	貞	呈	堤	定	帝	底	庭	廷	弟
	447X	悌	抵	挺	提	梯	汀	碇	禎	程	締	艇	訂	諦	蹄	逓	
	452X		邸	鄭	釘	鼎	泥	摘	擢	敵	滴	的	笛	適	鏑	溺	哲

次頁につづく

コードはすべて16進形式

テ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	453X	徹	撤	轍	迭	鉄	典	塡	天	展	店	添	纏	甜	貼	転	顛
	454X	点	伝	殿	澱	田	電										

(例えば，天のコードは4537と読みます．実際の使用は"&H"をつけて「&H4537」とします．)

ト		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	454X							兎	吐	堵	塗	妬	屠	徒	斗	杜	渡
	455X	登	菟	賭	途	都	鍍	砥	礪	努	度	土	奴	怒	倒	党	冬
	456X	凍	刀	唐	塔	塘	套	宕	島	嶋	悼	投	搭	東	桃	檮	棟
	457X	盗	淘	湯	濤	灯	燈	当	痘	禱	等	答	筒	糖	統	到	
	462X		董	蕩	藤	討	謄	豆	踏	逃	透	鐙	陶	頭	騰	闘	働
	463X	動	同	堂	導	憧	撞	洞	瞳	童	胴	萄	道	銅	峠	鴇	匿
	464X	得	徳	瀆	特	督	禿	篤	毒	独	読	栃	橡	凸	突	椴	届
	465X	鳶	苫	寅	酉	瀞	噸	屯	惇	敦	沌	豚	遁	頓	呑	曇	鈍

ナ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	466X	奈	那	内	乍	凪	薙	謎	灘	捺	鍋	楢	馴	縄	畷	南	楠
	467X	軟	難	汝													

ニ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	467X				二	尼	弐	邇	匂	賑	肉	虹	廿	日	乳	入	
	472X		如	尿	韮	任	妊	忍	認								

ヌ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	472X									濡							

ネ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	472X										禰	祢	寧	葱	猫	熱	年
	473X	念	捻	撚	燃	粘											

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ノ	473X						乃	廼	之	埜	嚢	悩	濃	納	能	脳	膿
	474X	農	覗	蚤													

（例えば，能のコードは473Dと読みます．実際の使用は“&H”をつけて「&H473D」とします）

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ハ	474X				巴	把	播	覇	杷	波	派	琶	破	婆	罵	芭	馬
	475X	俳	廃	拝	排	敗	杯	盃	牌	背	肺	輩	配	倍	培	媒	梅
	476X	楳	煤	狽	買	売	賠	陪	這	蠅	秤	矧	萩	伯	剥	博	拍
	477X	柏	泊	白	箔	粕	舶	薄	迫	曝	漠	爆	縛	莫	駁	麦	
	482X		函	箱	硲	箸	肇	筈	櫨	幡	肌	畑	畠	八	鉢	潑	発
	483X	醱	髪	伐	罰	抜	筏	閥	鳩	噺	塙	蛤	隼	伴	判	半	反
	484X	叛	帆	搬	斑	板	氾	汎	版	犯	班	畔	繁	般	藩	販	範
	485X	釆	煩	頒	飯	挽	晩	番	盤	磐	蕃	蛮					

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ヒ	485X												匪	卑	否	妃	庇
	486X	彼	悲	扉	批	披	斐	比	泌	疲	皮	碑	秘	緋	罷	肥	被
	487X	誹	費	避	非	飛	樋	簸	備	尾	微	枇	毘	琵	眉	美	
	492X		鼻	柊	稗	匹	疋	髭	彦	膝	菱	肘	弼	必	畢	筆	逼
	493X	檜	姫	媛	紐	百	謬	俵	彪	標	氷	漂	瓢	票	表	評	豹
	494X	廟	描	病	秒	苗	錨	鋲	蒜	蛭	鰭	品	彬	斌	浜	瀕	貧
	495X	賓	頻	敏	瓶												

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
フ	495X					不	付	埠	夫	婦	富	冨	布	府	怖	扶	敷
	496X	斧	普	浮	父	符	腐	膚	芙	譜	負	賦	赴	阜	附	侮	撫
	497X	武	舞	葡	蕪	部	封	楓	風	葺	蕗	伏	副	復	幅	服	
	4A2X		福	腹	複	覆	淵	弗	払	沸	仏	物	鮒	分	吻	噴	墳
	4A3X	憤	扮	焚	奮	粉	糞	紛	雰	文	聞						

コードはすべて16進形式

ヘ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4A3X											丙	併	兵	塀	幣	平
4A4X	弊	柄	並	蔽	閉	陛	米	頁	僻	壁	癖	碧	別	瞥	蔑	箆
4A5X	偏	変	片	篇	編	辺	返	遍	便	勉	娩	弁	鞭			

（例えば，平のコードは4A3Fと読みます．実際の使用は"&H"をつけて「&H4A3F」とします）

ホ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4A5X														保	舗	鋪
4A6X	圃	捕	歩	甫	補	輔	穂	募	墓	慕	戊	暮	母	簿	菩	倣
4A7X	俸	包	呆	報	奉	宝	峰	峯	崩	庖	抱	捧	放	方	朋	
4B2X		法	泡	烹	砲	縫	胞	芳	萌	蓬	蜂	褒	訪	豊	邦	鋒
4B3X	飽	鳳	鵬	乏	亡	傍	剖	坊	妨	帽	忘	忙	房	暴	望	某
4B4X	棒	冒	紡	肪	膨	謀	貌	貿	鉾	防	吠	頬	北	僕	卜	墨
4B5X	撲	朴	牧	睦	穆	釦	勃	没	殆	堀	幌	奔	本	翻	凡	盆

マ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4B6X	摩	磨	魔	麻	埋	妹	昧	枚	毎	哩	槙	幕	膜	枕	鮪	柾
4B7X	鱒	桝	亦	俣	又	抹	末	沫	迄	儘	繭	麿	万	慢	満	
4C2X		漫	蔓													

ミ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C2X				味	未	魅	巳	箕	岬	密	蜜	湊	蓑	稔	脈	妙
4C3X	粍	民	眠													

ム

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C3X				務	夢	無	牟	矛	霧	鵡	椋	婿	娘			

メ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C3X														冥	名	命
4C4X	明	盟	迷	銘	鳴	姪	牝	滅	免	棉	綿	緬	面	麺		

ル		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4E5X													瑠	塁	涙	累
	4E6X	類															

（例えば，類のコードは 4 E60 と読みます．実際の使用は "&H" をつけて「&H4 E60」とします）

レ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4E6X		令	伶	例	冷	励	嶺	怜	玲	礼	苓	鈴	隷	零	霊	麗
	4E7X	齢	暦	歴	列	劣	烈	裂	廉	恋	憐	漣	煉	簾	練	聯	
	4F2X		蓮	連	錬												

ロ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4F2X					呂	魯	櫓	炉	賂	路	露	労	婁	廊	弄	朗
	4F3X	楼	榔	浪	漏	牢	狼	籠	老	聾	蝋	郎	六	麓	禄	肋	録
	4F4X	論															

ワ		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	4F4X		倭	和	話	歪	賄	脇	惑	枠	鷲	亙	亘	鰐	詫	藁	蕨
	4F5X	椀	湾	碗	腕												

コードはすべて16進形式

モ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C4X															摸	模
4C5X	茂	妄	孟	毛	猛	盲	網	耗	蒙	儲	木	黙	目	杢	勿	餅
4C6X	尤	戻	籾	貰	問	悶	紋	門	匁							

（例えば，門のコードは4C67と読みます．実際の使用は"&H"をつけて「&H4C67」とします）

ヤ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C6X										也	冶	夜	爺	耶	野	弥
4C7X	矢	厄	役	約	薬	訳	躍	靖	柳	薮	鑓					

ユ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C7X												愉	愈	油	癒	
4D2X		諭	輸	唯	佑	優	勇	友	宥	幽	悠	憂	揖	有	柚	湧
4D3X	涌	猶	猷	由	祐	裕	誘	遊	邑	郵	雄	融	夕			

ヨ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4D3X														予	余	与
4D4X	誉	輿	預	傭	幼	妖	容	庸	揚	揺	擁	曜	楊	様	洋	溶
4D5X	熔	用	窯	羊	耀	葉	蓉	要	謡	踊	遥	陽	養	慾	抑	欲
4D6X	沃	浴	翌	翼	淀											

ラ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4D6X						羅	螺	裸	来	莱	頼	雷	洛	絡	落	酪
4D7X	乱	卵	嵐	欄	濫	藍	蘭	覧								

リ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4D7X									利	吏	履	李	梨	理	璃	
4E2X		痢	裏	裡	里	離	陸	律	率	立	葎	掠	略	劉	流	溜
4E3X	琉	留	硫	粒	隆	竜	龍	侶	慮	旅	虜	了	亮	僚	両	凌
4E4X	寮	料	梁	涼	猟	療	瞭	稜	糧	良	諒	遼	量	陵	領	力
4E5X	緑	倫	厘	林	淋	燐	琳	臨	輪	隣	鱗	麟				

付録5　F-BASIC エラーリスト

プログラムを修正する上で，修正箇所を明確にするため，F-BASIC の出力するエラーの意味とその発生原因の主な例をあげておきます．

ERR コード　01：Next Without For

意味：FOR 文がないのに NEXT 文が出てきた．

原因：Program ミス　FOR～NEXT が正しく使われていない．

チェックポイント

：FOR の数と NEXT の数とが合っているか？

：FOR～NEXT ループ内で GOTO 又は GOSUB などがないか．

　ある場合分岐先のプログラムもチェックすること．

悪い例

```
   :                         FOR I=1  TO 20
 70 GOTO(GOSUB)200    200 :
   :                         NEXT I
   :                         GOTO80(RETURN)
                             :
```

ERR コード　02：Syntax Error

意味：プログラムの文法（文の書き方）がまちがっている．

原因：Program ミス　プログラム中に F-BASIC にない命令文がある．

　または，命令が正規の型式以外で使用されている．

チェックポイント

：関数が代入文の左辺にきたり，命令として使用されていないか．

：変数名は英字で始まっているか．

：変数名に F-BASIC の命令（キーワード）が使用されていないか．

：ERR, ERL, CSRLIN の予約変数に値を代入しようとしていないか．

：行番号が，許される範囲内にあるか．（63999 までの正の整数）

：IF 文中で THEN がないのに ELSE が使用されていないか．

：関数の引数の数が多くないか．

：COLOR 文でパラメータが多くないか．例 COLOR　7, 1, 3

：LOCATE でパラメータが少ない．例 LOCATE

：DEF　FN で定義した関数の使い方がまちがっていないか．

ERRコード　03：Return Without Gosub

意味：GOSUBによってサブルーチンに飛んだわけでないのにRETURN文に出会った.

原因：Programミス　GOSUBを使用しないでサブルーチンへ分岐した.

チェックポイント

：サブルーチンへGOTO文で分岐していないか.

：メインルーチンはきちんと終るようになっているか.

：メインルーチンを終ると自動的にサブルーチンを実行してしまうようになってないか.

悪い例　メインルーチン　　　　　サブルーチン

```
        :                  *SUB1  210  :
        :                         250  RETURN
     GOSUB  210            *SUB2  290  :  ETURN
        :                              :
     GOSUB  290                   350  RETURN
        :
  200   :
```

上の例は，メインプログラムが，200行で終っているのにEND文がないため，350行でRETURNしてくると＊SUB1を実行してしまい250行のRETURNに出会う．200行の後に，END文を入れてやるとOKとなる．

ERRコード　04：Out of Data

意味：READ文で読まれるデータ（DATA文で書かれているデータ）が足りない.

原因：Programミス　READ文の実行回数またはREADする変数の数よりDATAの数が少ない.

チェックポイント

：DATAの型式や数は合っているか.

：RESTOREの使い方は合っているか.

：RESTOREを使用していないのに，同じDATAをリードしようとしていないか.

ERRコード　05：Illegal Function Call

意味：関数やステートメント（命令）の呼び方が間違っている.

原因：Programミス　パラメータの設定または値が違う.

チェックポイント

：パラメータの範囲は合っているか.

：パラメータの使用する数の形式は合っているか.（整数のパラメータに負の数が使用されていないか等）

：宣言されていない配列や使用されていない変数を用いていないか．

(GET@, PUT@, SWAP, VARPTR 等)

：指定された配列の大きさが足りない．

(GET@, PUT@)

：配列の添字の値が負になっていないか．

：PRINT　USING での桁指定が 24 桁をこえていないか．

ERR コード　06：Overflow

意味：入力された数値，代入される数値，演算結果などが許される範囲を越えている．

原因：Program ミス　変数型式（単精度，倍精度，整数）以上に演算結果の値が大きくなった．

チェックポイント

：整数演算で演算結果が −32768〜32767 の範囲になっているか．

：実数演算などの結果が，1.7014×10^{38}〜-1.7014×10^{38}の範囲にあるか．

：POKE, PEEK 文などの番地指定は合っているか．

(&H0000〜&HFFFF：16 進)(10 進 0〜65535 まで)

ERR コード　07：Out of Memory

意味：メモリが足りない．

原因：Program ミス　プログラムがメモリ容量より多い．

チェックポイント

：プログラムは収まっても，変数などの領域をとるとメモリが不足してしまわないか．

：DIM 文の配列が大きすぎないか．

：GOSUB, FOR〜NEXT 文が多すぎると BASIC のインタプリタがスタック領域をとるのでメモリ領域がへってしまう．

：PRINT　FRE(0) を実行してやり残りのメモリ領域を確認するとよい．

ERR コード　08：Undefined Line Number

意味：行番号が定義されていない．指定された行番号が存在しない．

原因：Program ミス　存在しない行番号をプログラム中に指定している．

チェックポイント

：GOTO, GOSUB, DELETE, LIST 等において，指定する行番号が，プログラム中に存在しているか．

ERR コード　09：Subscript Out of Range

意味：配列の添字が0からDIM文によって宣言された大きさの範囲内にない．

原因：Program ミス　配列の添字の値が大きすぎる．

チェックポイント

：FOR～NEXTにより配列の添字を変化させている場合，その値は宣言した添字以内か．

：配列の次元数（1次元配列に2次元指定をした）を誤っていないか．

：DIM文を使用しないで添字を10より大きな値を指定していないか．

ERR コード　10：Duplicate Definition

意味：同じ名前の配列または，ユーザ関数を二度宣言している．

原因：Program ミス　同じDIM文を2回以上実行しようとしている．

チェックポイント

：FOR～NEXT，GOTOのループ内で同じ名前の配列を使用していないか．（ユーザ関数を同じ名前で宣言していないか）

：サブルーチン内で同じ名前の配列を使用していないか．

（変数名は最初の16文字＋形宣言（%，!，#）で判断している．）

ERR コード　11：Division By Zero

意味：0による除算を実行しようとした．

原因：Program ミス　除算の除数が0である．

チェックポイント

：定義されていない変数を除数に使用していないか．

：関数tanでの引数が$\pi/2$になっていないか．

ERR コード　12：Illegal Direct

意味：直接モードでは使用できない命令を直接モードで実行しようとした．

原因：Operation ミス　間接モードのみの命令を直接モードで実行した．

(DEFFN, INPUT, LINE INPUT)

ERR コード　13：Type Mismatch

意味：変数の型があわない．

原因：Program ミス　数値を文字変数に代入しようとした．または，文字を数値変数に代入しようとした．

チェックポイント

：数値を文字変数に代入しようとしていないか．

：文字を数値変数に代入しようとしていないか．

：関数の引数の型はあっているか．

：FOR文での制御変数に倍精度実数を使用していないか．

：＊数値変数に文字を代入している例

INPUT, "アナタ　ノ　ナマエハ", NAME

：＊FOR文の制御変数に倍精度実数を使用した例

```
FOR  I#=1  TO  7
COLOR  I#:PRINT "FUJITSU"
NEXT  I#
```

ERRコード　14：Out of String Space

意味：文字領域が足りない．

原因：Programミス　プログラム内で使用する文字の量が多すぎる．

チェックポイント

：電源投入時は，BASICインタプリタで文字領域は最大300文字に設定されている．

：300以上設定する場合はCLEAR文により領域を確保しなければならない．

ERRコード　15：String Too Long

意味：文字が長すぎる．

原因：ProgramまたはOperationミス　文字変数に256文字以上入力しようとした．

チェックポイント

：文字変数等の演算の結果が256文字以上になっていないか．

：キーより，または入出力装置より1つの文字変数に256文字以上入力していないか．

：F-BASICでは1つの文字変数の最大長は255文字まで．

ERRコード　16：String Formula Too Complex

意味：文字式が複雑すぎる．

ERRコード　17：Can't Continue

意味：CONTコマンドによるプログラム続行ができない．

原因：OperationミスBreakキーまたはSTOP文でプログラム停止後，プログラムの変更等を

行なったため.

チェックポイント

：停止後プログラムの変更を行なうと CONT コマンドは，実行できない．また停止後 LIST コマンドまたは FILES コマンドを実行中，Break キーを押下後の CONT コマンドも実行できない．

ERR コード　18：Undefined User Function

意味：定義されていない関数が参照された．

原因：Program ミス　DEF　FN で定義されていない関数を参照した．

チェックポイント

：配列名の最初に FN を使用していないか．

ERR コード　19：NO Resume

意味：エラー処理ルーチンに RESUME 文がない．

原因：Program ミス　エラー処理ルーチンに RESUME 文がない．

チェックポイント

：エラー処理ルーチンは，RESUME{ 行番号，NEXT }，END のいずれかで終了しなければならない．

ERR コード　20：Resume Without Error

意味：エラーが起きていないのに RESUME 文を実行しようとした．

原因：Program ミス

チェックポイント

：GOTO や GOSUB 等でエラー処理ルーチンへ分岐していないか．

：メインルーチンの処理後，エラー処理ルーチンを実行していないか．

（メインルーチンと，エラー処理ルーチンは END によって区別してやるとよい）

ERR コード　21：Unprintable Error

意味：エラーコードの定義されていないエラーを出そうとした．

原因：Program ミス

チェックポイント

：エラーメッセージ表にないエラーコードまたは，定義されていないエラーを発生させていないか．

：DISKがついていないのにDISKのエラーを発生させていないか．

ERRコード　22：Missing Operand

意味：必要なオペランドが欠けている．

原因：Programまたは Operationミス

チェックポイント

：WIDTH, COLOR, LOCATE等のパラメータが欠けていないか．

例　WIDTH　80，␣　　␣の部分のパラメータがない．代入文の右辺がないのではないか．

```
COLOR␣
LOCATE 10, ␣
A=␣
```

ERRコード　23：For Without Next

意味：FOR〜NEXTの対応が正しくない．

原因：（エラーコード23は実際にはエラーが発生しない）

FOR文に対しNEXTがなくてもエラーにならない．

ERRコード　24：While Without Wend

意味：WHILE文に対応するWEND文がない．

原因：Programミス

チェックポイント

：下記のプログラムのようになっていないか．

（2つのWHILE文に対し1つのWENDで対応させようとしていないか．）

```
10 B=10:INPUT "A ", A
20 WHILE A<>B
30 A=A+1:PRINT A
40 GOTO 70
50 WEND
60 END
70 WHILE A<>B
80 C=C+1:PRINT C
90 GOTO 50
```

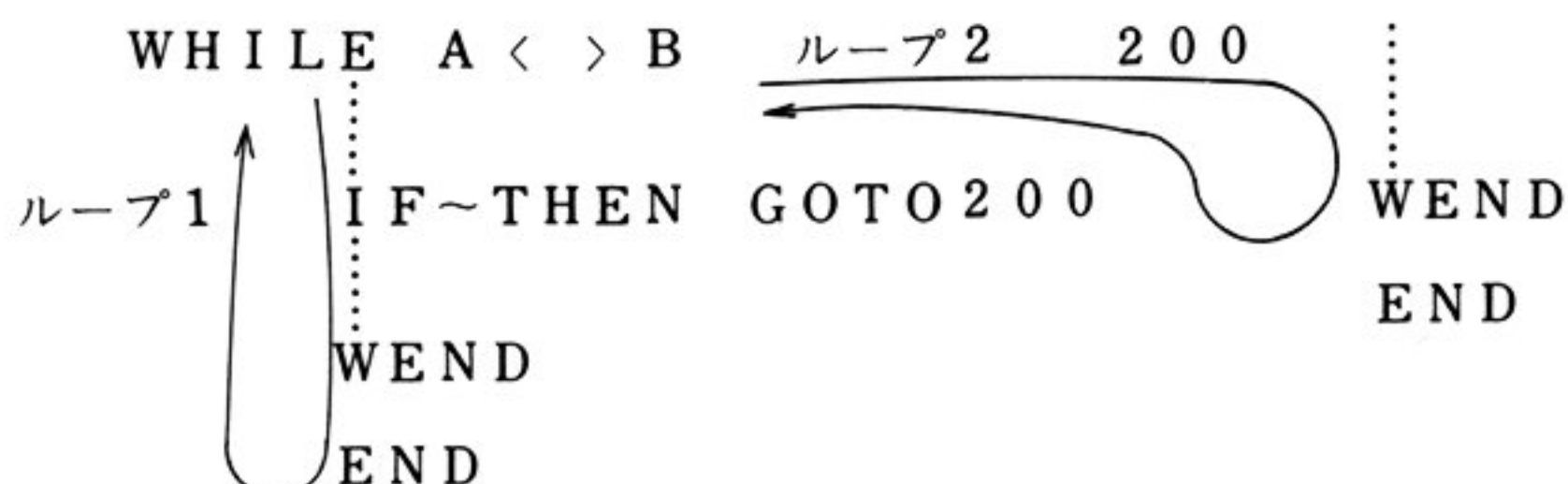
ERR コード 25：Wend Without While

意味：WEND 文に対応する WHILE 文がない.

原因：Program ミス　WHILE 文がないのに WEND 文を実行しようとした.

チェックポイント

：下記のプログラム構成になっていないか.

```
            WHILE A< >B      ループ2   200
                  :                        :
ループ1          IF~THEN  GOTO200          WEND
                  :                        END
                 WEND
                 END
```

F-BASIC エラーメッセージ(I/O 系)

ERR コード 26：Bubble Full（このエラーは，FM-7 には存在しません）

意味：バブルカセットがいっぱいでデータの登録ができない.

原因：Operation ミス　バブルの容量が少ない状態で大容量のデータを記録しようとした.

チェックポイント

：データを記録する前に，FILES "BUB□：" で容量を確認する.

（□：ユニット No.）

```
FILES "BUB0:"
TS          0BS11
ONLINE      1AS  3
GOLF        0BS726
                ↑ 使用ページ数
283 Pages Free
↑ 残りページ
```

残り容量　283×32バイト＝9056バイト

ERR コード 50：Bad File Number

意味：ファイル番号が誤っている.

原因：Program ミス

チェックポイント

：指定したファイル番号が 0 から BASIC の起動時に定めた上限までの間にないのではないか.

：シーケンシャルファイルでのファイル操作に，ファイル番号 0 を使っていないか.

ERR　51：Bad File Mode

意味：ファイルモードが誤っている.

原因：Program ミス　入力（出力）でオープンしたファイルに出力（入力）しようとした.

チェックポイント

：オープン文の入力，出力に対し入出力命令が対応しているか.

例

```
OPEN "I", 1, "0:"          }入力でオープンしたのに
PRINT#1, "FUJITSU"         }出力命令を実行した.
Bad File Mode
Ready
■
```

：オープン文の入力，出力の出力の指定を誤っていないか.

```
OPEN "O", 1, "0:"
OPEN "i", 1, "1:"
      ↑
      └────かならず大文字で記入すること.
```

ERR コード　52：File Already Open

意味：指定したファイルはすでにオープンされている.

原因：Program ミス　同一のファイルを2回オープンしようとした.

チェックポイント

：すでに使用中のファイル番号を指定してファイルをオープンしようとしていないか.

：ファイルがオープンされている状態でKILLを実行していないか.

(KILLはファイルがCLOSEされていないとできない.)

ERR コード　53：Device I/O Error

意味：使用しているI/Oにエラーが発生した.

原因：使用媒体不良

チェックポイント

：各I/OによりDevice I/O Errorの条件が違いますので注意して下さい.

：カセットレコーダによるDevice I/O Errorは，レコーダとの相性がありますのでいちがいに，機器が悪いとはいえませんので注意して下さい.

：また，カセットのSAVE時や，プリンタ，漢字，アナログポート出力時，入力時は，Device I/Oエラーは出力されません.

：バブルホルダユニットで，バブルカセットをライトプロテクト状態にしておき，書込みを行なうと，Device I/O エラーが発生しますので注意して下さい.（FM-7を除く）

ERR コード　54：Input Past End

意味：ファイルのデータを全て読んだ後に，入力命令（INPUT #，LINE INPUT #）を実行しようとした.

原因：Program ミス　EOF（End of File）を越えて入力命令を実行した.

チェックポイント

：EOF を越えて INPUT #，命令等を実行していないか.

：GET#，で指定したレコード番号がファイルの大きさを越えていないか.

：シーケンシャルファイルでは，INPUT # 等の命令の前に EOF のチェックを行なうこと．ランダムファイルでは LOF でレコード番号の最大を調べておきそれより大きな番号を指定しないこと.

例

```
10  OPEN "I", 1, "TEST"
20  PRINT "データファイルノリード"
30  IF EOF(1) THEN PRINT "リードオワリ":END
40  INPUT#1, A$
50  PRINT A$
60  GO TO 30
```

ERR コード　55：Bad File Descriptor

意味：ファイルディスクリプタの記述に誤りがある.

原因：Program ミス　ファイル名に，ファイル名に使用してはいけない記号（：，〃，左カッコ，右カッコ）が含まれている.

チェックポイント

：ファイル名が指定されていないのではないか.

：ファイル名が長すぎるのではないか.（最大8文字）

```
OPEN "0", 1, "デバイス名：ファイル名"
                         ↑
                         └ここが誤っている.
```

ERR コード　56：Direct Statement In File

意味：プログラムロード時行番号なしの命令がある.

原因：Operation, Program ミス　このエラーは，アスキー形式のプログラムをロードしているときのみ出力される．

チェックポイント

：単なるデータファイルを，まちがって LOAD していないか．

ERR コード　57：File Not Open

意味：入力，出力命令を実行したとき，指定したファイル番号に対するファイルがオープンされていない．

原因：Program ミス

チェックポイント

：オープン命令のファイル番号と入力（または出力）命令のファイル番号が一致してないのではないか．

ERR コード　58：Bad Data In File

意味：File 上のデータ形式と入力命令の変数の形式とが異なっている．

原因：Program ミス

チェックポイント

：DATA と出力した時の変数形式と入力命令の変数形式が一致しているか．

例

出力プログラム

```
 10 OPEN "O", 1, "DATA"
 20 INPUT "DATAハ", A!
 30 PRINT#1, A!
 40 CLOSE:END
```

入力プログラム

```
100 OPEN "I", 2, "DATA"
110 IF EOF(2) THEN END
120 INPUT#2, A%:PRINT A%
130 GOTO 110
Bad Data In File in 120
Ready
■
```

変数の形式が違う．

ERR コード　59：Device In Use

意味：使用中の入出力装置をオープンしようとした．

原因：Program, Operation ミス

チェックポイント

：使用した入出力装置を CLOSE しないで，またオープンしようとしていないか．

例

```
10 ,Device In Use
20 OPEN "O", 1, "CAS0:TR"
30 PRINT#1, "Device In Use"
40 OPEN "O", 2, "CAS0:TO"
50 PRINT#1, "Device In Use"
60 CLOSE#1, #2

   Device In Use In 40
   Ready
   ■
```

ERR コード　60：Device Unavailable

意味：入出力装置が入出力可能な状態にない．

原因：Program, Operation ミスおよびハードエラーも考えられる．

入出力しようとする目的の装置が接続されていない，または使用できる状態でない．

チェックポイント

：このエラーは OPEN 文でのエラーでデバイス名の違い等が考えられる．

下記の点に注意して下さい．

```
Open "O", 1, "CAS0:TR"
              ↑
              └──── デバイス名は，かならず大文字で書くこと．

Open "O", 2, "BUB0:TR"
                 ↑↑
                 │└──── コロンは忘れずに付けること．
                 └───── ユニット NO, "0" と "O" を間違えないこと．
```

注意：バブルカセットでイニシャライズ（BUBINI）をしていない，またはディレクトリ部分をこわした場合は，このエラーになります．（FM-7 を除く）

ERR コード　61：Buffer Overflow

意味：入出力装置のバッファが処理する前にいっぱいになってしまった．

原因：Program ミス

チェックポイント

：このエラーは，RS-232C(Open "I" または "O"，1，"COM0：")を使用時に，発生する場合があります．

データの伝送速度が速すぎてF-BASICのスピードでは，データをバッファから取り込むのが遅すぎるために起ります．ボーレートを遅くして下さい．

：転送速度は充分受信できる速度であるが，全体のプログラムの処理速度が遅すぎるときにも発生します．

ERRコード　62：Protected Program

意味：保護されているプログラムに書込み修正を行なおうとした．

ERRコード　63：File Not Found

意味：指定されたファイルが存在しない．

原因：Operation，　Programミス　存在しないファイルを指定した．

チェックポイント

：ファイル名は，大文字，小文字の区別が有るがそれを間違えてはいないか．

：ドライブナンバー等の指定は正しいか．

ERRコード　64：File Already Exists

意味：指定したファイルは，すでに存在する．

原因：Programミス，すでにあるファイル名で，再度データを登録しようとした．

チェックポイント

：OPEN文で入力指定を行なおうとするファイルを，出力指定をしていないか．

ERRコード　65：Directory Full

意味：ディレクトリ領域がいっぱいで，新たなファイルが登録できない．

原因：Operationミス

チェックポイント

：ファイルが登録できるかどうかは，Filesにより知ることができる．

なお，1フロッピィディスクに登録できるファイルの最大量は，152までである．

ERRコード 66：Too Many Open Disk Files

意味：確保されているファイルの数をこえてオープンしようとした．

原因：Program ミス

チェックポイント

：下記のようになっていないか．

初期設定（How many Disk Files）で2を設定した場合．

例

```
10 OPEN "O", 1, "P"
20 PRINT#1, "PPPPP"
30 OPEN "O", 2, "A"
40 PRINT#2, "AAAAAA"
50 OPEN "O", 3, "R"
60 PRINT#3, "RRRRR"
Too many Open Disk Files in 50
```

ERRコード 67：Disk Full

意味：ディスクがいっぱいでデータの登録ができない．

原因：Program ミス 1フロッピィディスクの空白残量（容量）より大きなデータを登録しようとした．

チェックポイント

：残りクラスタ（容量）以上のデータを登録しようとしていないか．

ERRコード 68：Field Overflow

意味：フィールドの長さが256バイトを越えている．

原因：Program ミス FIELD文の指定ミス．（ランダムファイル使用時）

チェックポイント

：FIELD文のフィールドが256バイトを越えていないか．

```
OPEN "R", 1, "RAND"
FIELD#1, 20 AS S$, 100 AS K$, 10 AS B$,
128 AS E$
    20+100+10+128=258>256バイト
```

ERRコード 69：String Not Fielded

意味：FIELD文で指定した文字変数以外をLSET，RSET文で代入しようとした．

原因：Programミス

チェックポイント

：例

```
FIELD#1, 10AS Fie$
LSET A$=Fie$
```

（A$ ← FIELD文で指定されていないのでエラーになる．）

ERRコード　70：Bad Record Number

意味：指定されたレコード番号は，存在しない．

ERRコード　71：Bad File Structure

意味：ファイルの構成に誤りがある．

原因：Operation, Program，ハードエラー，ディレクトリ，FATがこわれている．

チェックポイント

：F-BASICのDISKコードが書込まれていないディスクを使用していないか．

：プログラムでディレクトリ，FATの部分に記録していないか．

：FORMATしていないフロッピィディスクなどを使用していないか．

ERRコード　72：Drive Not Ready

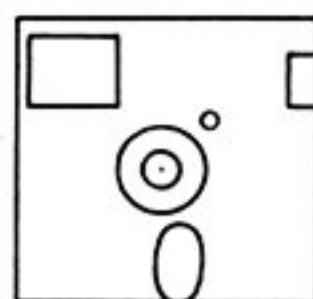
意味：指定したフロッピィユニットが，使用可能状態にない．

原因：Operation，ハードエラー

チェックポイント

：フロッピィディスクユニットのドライブカバーが開いていないか．

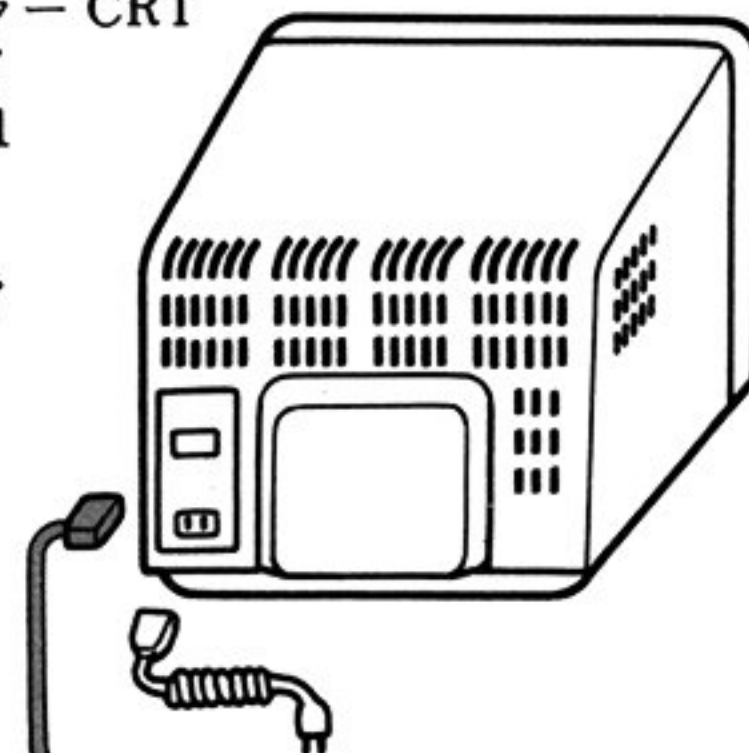
ERRコード　73：Disk Write Protected

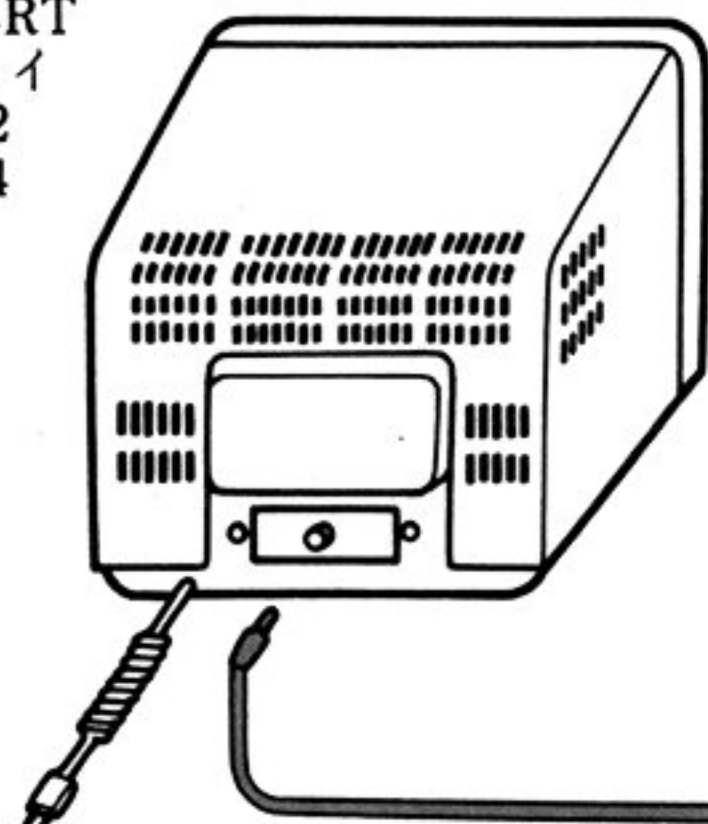
意味：フロッピィディスクが書込み禁止されている．

ライトプロテクトノッチ
ここにシールがはられていると，記録できません．

ミニフロッピィディスク

FUJITSU MICRO 7 機器接続図

高解像度カラー CRT
ディスプレイ
MB27301

カラー CRT
ディスプレイ
MB27303
MB27305

カラー CRT ケーブル
MB26512
(MB26501)

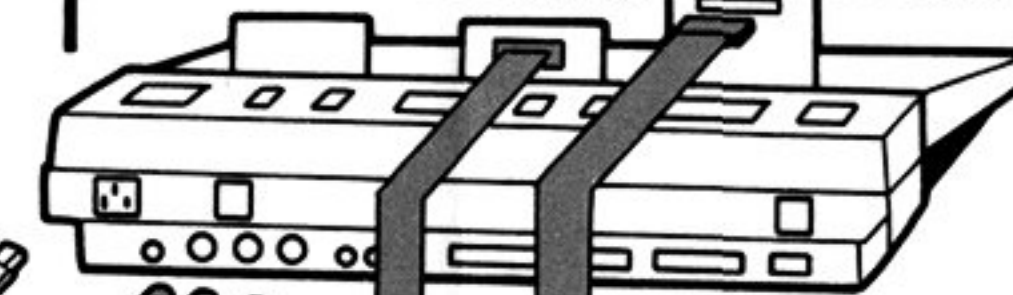
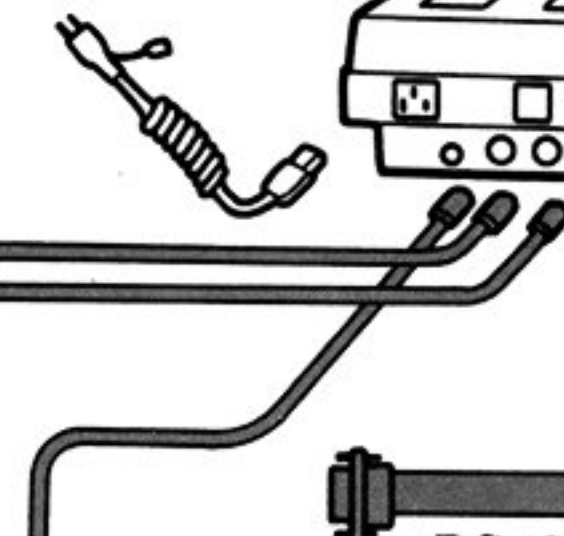
グリーン CRT
ディスプレイ
MB27302
MB27304

グリーン CRT ケーブル
MB26503

FUJITSU MICRO 7 本体
MB25010

Z80 カード
MB28021

RS-232C
インタフェースカード
MB22406

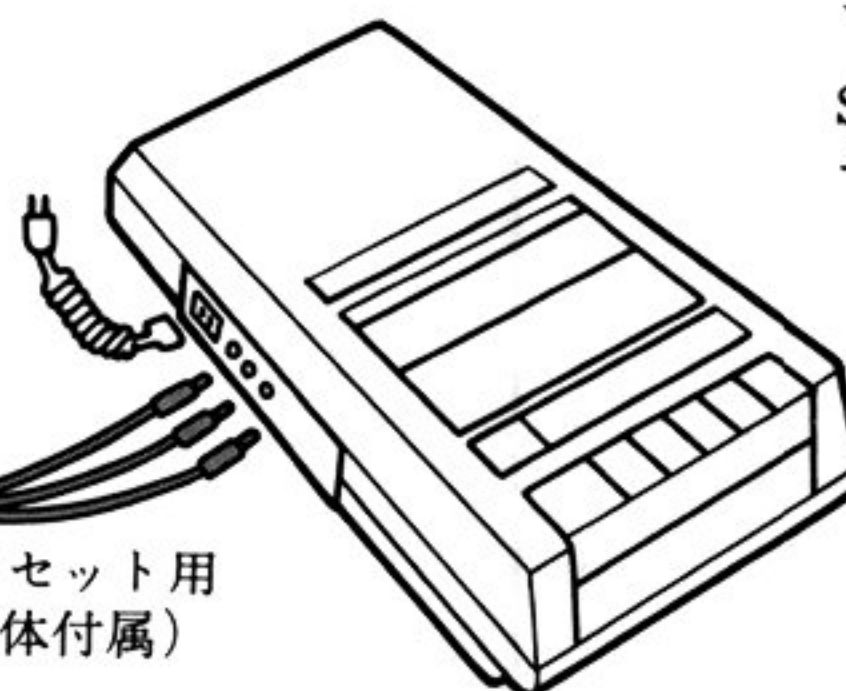
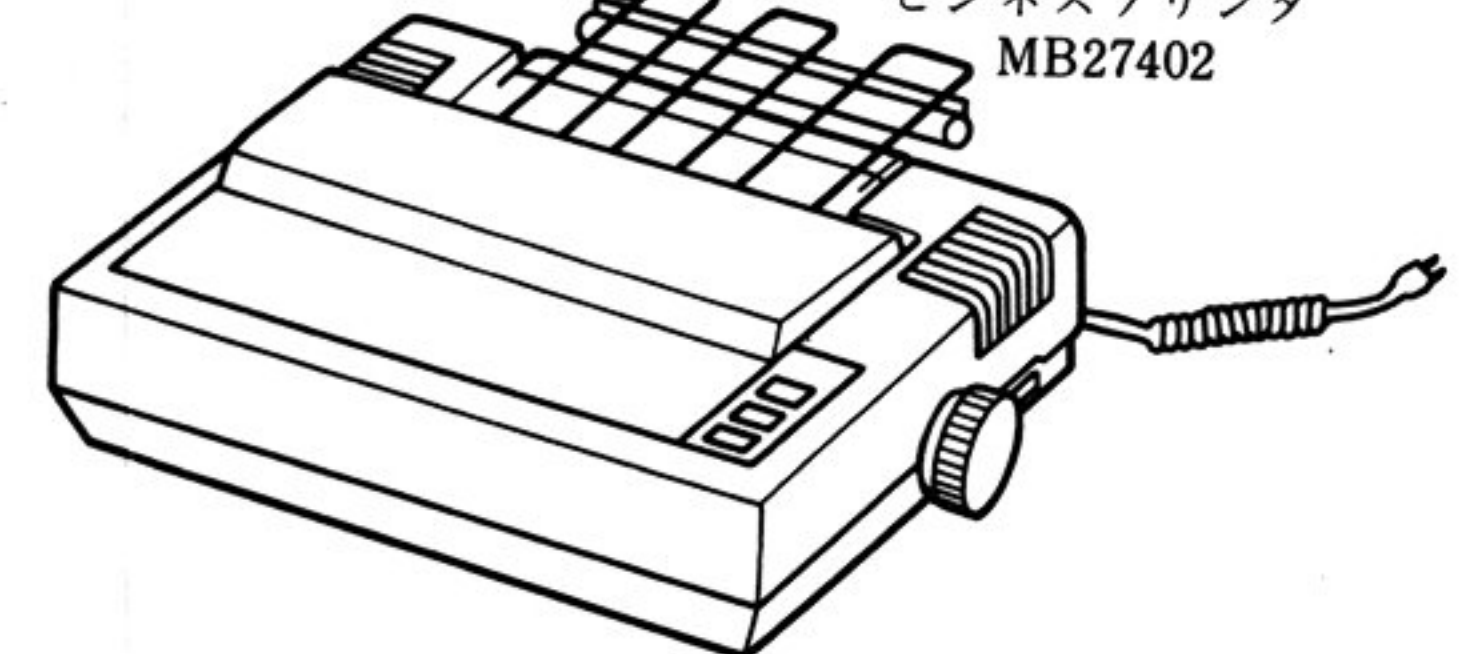
ミニフロッピィイン
タフェースカード
MB22407

(漢字 ROM カード
MB22405)

RS-232C
ケーブル
MB26505

オーディオカセット用
ケーブル(本体付属)

FM データレコーダ
MB27501

ドットプリンタ
MB27404

シリアルドットプリンタ
MB27403
(MB27401)
ビジネスプリンタ
MB27402

プリンタケーブル
MB26514

ミニフロッピィディスクユニット
MB27601
S タイプミニ FD
ユニット
MB27605

薄型ミニ FD ユニット
MB27607
増設用ミニ FD ユニット
MB27608